Министерство образования и науки Российской Федерации
ФГБОУ ВО «Уральский государственный педагогический университет»
Институт математики, информатики и информационных технологий
Кафедра высшей математики

Использование прикладного пакета GAP для описания решеток подалгебр моногенных трехмерных алгебр над полем GF(2)

Выпускная квалификационная работа

Квалификационная работа
допущена к защите
Зав. кафедрой

_____ _____
дата подпись

Исполнитель:
Васильев Степан Алексеевич,
обучающийся БП-41 группы

_____
подпись

Руководитель ОПОП:

_____
подпись

Научный руководитель:
Коробков С.С.,
к.ф.-м.н., доцент

_____
подпись

Екатеринбург 2016

Оглавление

Введение

Пусть $A = M_3(GF(2))$ – алгебра квадратных матриц порядка три над полем из двух элементов. **Основным объектом** исследования является класс моногенных, то есть однопорожденных, подалгебр алгебры A.

Интерес к матричным алгебрам во многом определяется той ролью, которую играют эти алгебры в теории ассоциативных алгебр. Уникальность алгебры матриц подтверждается и тем, что матричная алгебра определяется своей решеткой подалгебр. Как доказал D. Barnes [1] любая конечномерная алгебра над конечным полем F, решеточно изоморфная алгебре матриц $M_n(F)$, где $n>1$, изоморфна алгебре $M_n(F)$. Этот факт говорит о том, что и решетка подалгебр матричной алгебры сама является интересным объектом для глубокого изучения. Интерес к ней объясняется и рядом других причин. Одной из них является то, что эта решетка является богатым источником решеток подалгебр конечномерных алгебр ввиду известной теоремы о вложимости конечномерной алгебры в алгебру матриц. Однако, с увеличением порядка матриц количество подалгебр в алгебре $M_n(F)$ резко возрастает. Как оказалось, при $n=2$ и $F = GF(2)$ оно равно 28, а при $n=3$ и $F = GF(2)$ оно равно уже 2102. Количество типов решеток подалгебр соответственно возрастает с 5 до 30. Поэтому ясно, что случай, $n = 3$ – последний случай, когда еще можно проводить исследования вручную.

С появлением удобного компьютерного пакета для вычислений в самой алгебре $M_n(F)$ вновь усилился интерес к исследованию ее решетки подалгебр. Таким пакетом является система компьютерной алгебры **GAP** (Groups, Algorithms and Programming), которая разрабатывается уже более двадцати лет. Сегодня **GAP** является уникальным всемирным проектом, над которым работают специалисты-математики и представители других наук из различных уголков планеты. **GAP** — это свободно распространяемая открытая кроссплатформенная система, которая постоянно дополняется и расширяется. В последней версии **GAP 4.8.3** (март, 2016), которая доступна

на сайте www.gap-system.org, имеется множество удобных функций для исследования во многих разделах алгебры, что делает **GAP** чрезвычайно полезным инструментом для каждого исследователя, изучающего алгебраические объекты. **GAP** является свободно распространяемой, открытой и расширяемой системой.

Главной целью исследования является классификация моногенных алгебр с точностью до изоморфизма, а также описание решеток подалгебр таких алгебр в алгебре *A.* В качестве основного средства для достижения цели использовалась система компьютерной алгебры GAP. Достижение цели осуществлялось путем решения следующих **задач**:

1. Разработка алгоритмов и программ для нахождения всех моногенных подалгебр.
2. Выявление алгебраических свойств, необходимых и достаточных для классификации с точностью до изоморфизма.
3. Исследование порядков подалгебр порожденных двумя элементами.
4. Нахождение решеток подалгебр моногенных алгебр.

Сформулируем необходимые в дальнейшем определения.

Определение 1. Пусть S – подалгебра порядка 2^n алгебры $M_3(GF(2))$. Назовем упорядоченную последовательность $(m_0,m_1, \dots ,m_n)$ *типом* решетки подалгебр алгебры S, если m_i – число подалгебр в S порядка 2^i.

Определение 2. Решетки, диаграммы которых представлены на рисунке 1, будем называть соответственно *решетками* I, II, III, IV, V, VI.

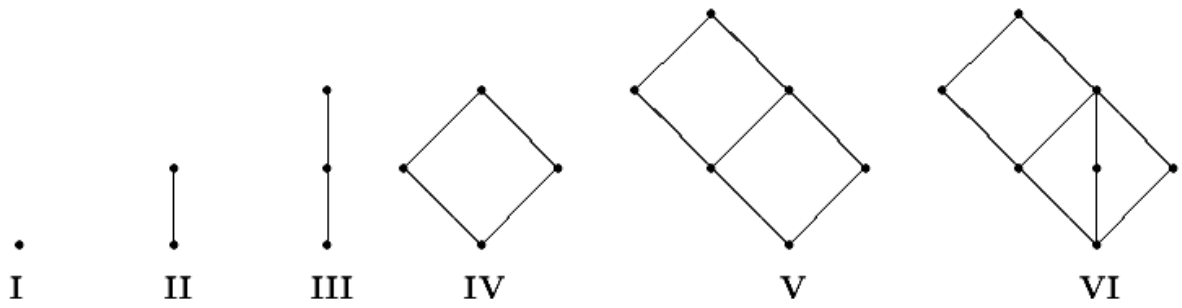


Рис. 1.

Работа состоит из введения, трех глав, библиографического списка и приложения.

Теоретические основы работы изложены в первой главе. В ней приводятся основы теории ассоциативных алгебр и теории решеток.

Во второй главе рассматривается прикладной пакет GAP, его основные команды, используемые в работе.

В третьей главе находятся все моногенные подалгебры алгебры $M_3(GF(2))$ и производится их полная классификация с точностью до изоморфизма. Находятся решетки подалгебр и их типы для всех моногенных подалгебр из $M_3(GF(2))$.

Основные результаты работы приведены в таблице № 1.

Таблица № 1

№	Тип моногенной алгебры	Кол – во подалгебр	Решетка подалгебр	Тип решетки подалгебр
1	Нулевая алгебра	1	I	(1)
Моногенные подалгебры второго порядка				
2	$\langle r\rangle$, где $r^2 = 0$	21	II	(1, 1)
3	$\langle e\rangle$, где $e^2 = e \neq 0$	57	II	(1, 1)
	Итого:	**78**		**1**
Моногенные подалгебры четвертого порядка				
4	$\langle r\rangle$, где $r^3 = 0$, $r^2 \neq 0$	21	III	(1, 1, 1)
5	$\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=0$	84	IV	(1, 2, 1)
6	$\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=r$	105	IV	(1, 2, 1)
7	$GF(2^2)$	28	III	(1, 1, 1)
	Итого:	**238**		**2**
Моногенные подалгебры восьмого порядка				
8	$GF(2^3)$	8	III	(1, 1, 0, 1)
9	$\langle r\rangle \oplus \langle e\rangle$, где $r^3=0$, $r^2 \neq 0$, e – единица	21	V	(1, 2, 2, 1)
10	$GF(2^2) \oplus \langle e\rangle$, где e – единица	28	VI	(1, 3, 2, 1)
	Итого:	**57**		**3**
	Итого:	**374**		**7**

Сформулируем основные следствия, вытекающие из полученных результатов:

Следствие 1. Любая моногенная подалгебра в $M_3(GF(2))$ содержит не более двух максимальных подалгебр.

Следствие 2. Если решетка подалгебр моногенной алгебры S в $M_3(GF(2))$ не является цепью, то S содержит не менее двух минимальных подалгебр.

Следствие 3. Каждая решетка подалгебр, кроме решетки типа III, определяется своим типом.

Следствие 4. Подалгебра S, изоморфная полю $GF(2^3)$, определяется в $M_3(GF(2))$ своим типом решетки.

Следствие 5. Подалгебра S, изоморфная любой из алгебр

1) $\langle r\rangle \oplus \langle e\rangle$, где $r^3 = 0$, $r^2 \neq 0$, e – единица;

2) $GF(2^2)\oplus\langle e\rangle$, где e – единица,

определяется в $M_3(GF(2))$ своей решеткой подалгебр.

Глава 1. Основные понятия и факты теории алгебр над полем и теории решеток

1.1. Определение алгебры матриц.

Определение 1. *Алгеброй над полем* P называется множество A, на котором определены две бинарные операции “+” (сложение) и “$\cdot$” (умножение), а также операция умножения элементов из P на элементы из A (то есть отображение $P \times A \to A$), удовлетворяющие следующим условиям:

1) $(A,+,\cdot)$ – кольцо;

2) $(A,+)$ – векторное пространство над полем P;

3) $\forall \alpha \in P \ \forall a,b \in A \ (\alpha a)b = \alpha(ab) = a(\alpha b)$.

Пример 1. Пусть $M_n(F) = \left\{ \| a_{ij} \|, a_{ij} \in F \right\}$ - множество всех квадратных матриц с коэффициентами из поля F. Ясно, что $M_n(F)$ – кольцо относительно операций сложения и умножения квадратных матриц. Если определить умножение элементов из F на элементы из $M_n(F)$ следующим образом:

$$\forall \beta \in F \ \ \forall \| a_{ij} \| \in M_n(F) \quad \beta \| a_{ij} \| = \| \beta a_{ij} \|,$$

то относительно такого умножения и сложения матриц множество $M_n(F)$ становится векторным пространством над полем F.

Пусть $a = \| a_{ij} \|, b = \| b_{ij} \| \in M_n(F) \ \ \alpha \in F$. Тогда

$$(\alpha a)b = \| \alpha a_{ij} \| \cdot \| b_{ij} \| = \| \sum \alpha a_{ij} b_{ij} \| = \left(\sum a_{ij} (\alpha b_{ij}) \right) = a(\alpha b) = \left(\alpha \left(\sum a_{ij} b_{ij} \right) \right) = \alpha(ab)$$

Следовательно, $M_n(F)$ – алгебра над полем F. Эта алгебра называется алгеброй матриц над полем F.

Определение 2. Пусть A – алгебра над полем F. Назовем алгебру A *конечномерной*, если A, как векторное пространство над полем F, конечномерно. При этом размерность векторного пространства A над P будем называть размерностью или рангом алгебры A.

Пример 2. Базис алгебры $M_n(F)$ образуют матричные единицы $E_{ij} = (e_{ij})$, где $e_{i,j} = 0$, если $i \neq j$ и $e_{i,j} = 1$, если $i = j$. Следовательно, $\dim M_n(F) = n^2$.

1.2. Подалгебры.

Определение 3. Подмножество S алгебры A над полем P назовем *подалгеброй* алгебры, если относительно операций, определенных в A, S само является алгеброй над полем P.

Теорема 1. (Признак подалгебры). *Непустое подмножество S алгебры A над полем P тогда и только тогда является подалгеброй в A, когда выполнены следующие условия:*

1) $\forall a,b \in S \quad a-b \in S$;

2) $\forall a,b \in S \quad a \cdot b \in S$;

3) $\forall \alpha \in P \ \forall a \in S \quad \alpha a \in S$.

Доказательство. Пусть S – подалгебра алгебры A. Тогда очевидно, что условия 1) – 3) выполнены. Обратно: пусть выполнены условия 1) – 3). Тогда из выполнимости условий 1) и 2) следует, что S – подкольцо кольца A, а из выполнимости условий 1) – 3) следует, что S – векторное подпространство пространства A. Условие 3) выполняется в S, так как оно выполняется в A. Таким образом, S - подалгебра алгебры A.

Предложение 1. Пусть A_i $(i \in I)$ – подалгебры алгебры A над полем P. Тогда $\bigcap_{i \in I} A_i$ – подалгебра алгебры A.

Доказательство. Пусть $B = \bigcap_{i \in I} A_i$. Так как каждая подалгебра A_i $(i \in I)$ содержит нулевой элемент алгебры A, то $B = \varnothing$. Воспользуемся признаком подалгебры. Пусть $a,b \in B$. Тогда $a,b \in A_i$ $(i \in I)$ и потому $a-b,\ ab \in A_i$ $(i \in I)$ и потому $\alpha a \in B$. Таким образом, B - подалгебра алгебры A.

Пусть M – непустое подмножество алгебры A и $\{A_i \ (i \in I)\}$ – множество всех подалгебр алгебры A, содержащих M. Тогда согласно предложению 1 $\bigcap_{i \in I} A_i$ - подалгебра алгебры A. Ясно, что подалгебра $\bigcap_{i \in I} A_i$ является

наименьшей из всех подалгебр, содержащих множество M. Обозначим эту подалгебру следующим образом: $\langle M \rangle$. Подалгебра $\langle M \rangle$ называется подалгеброй, порожденной множеством M, а само M называется множеством образующих алгебры A. В случае, когда множество M одноэлементно, например $M = \{a\}$, подалгебру $\langle \{a\} \rangle$ будем записывать в виде $\langle a \rangle$ и называть моногенной или однопорожденной подалгеброй. Если в алгебре A содержится такой элемент a, что $A = \langle a \rangle$, то будем называть A *моногенной* алгеброй.

Теорема 2. *Пусть A – алгебра над полем P и $a \in A$. Тогда*

$\langle a \rangle = \{\alpha_1 a + \alpha_2 a^2 + \cdots + \alpha_n a^n \mid n \in N \ \& \ a_1, ..., a_n \in P\}$.

Доказательство. Включение $\{\alpha_1 a + \alpha_2 a^2 + \cdots + \alpha_n a^n \mid n \in N \ \& \ a_1, ..., a_n \in P\} \subseteq \langle a \rangle$ очевидно. Применяя признак подалгебры, легко убедиться в том, что подмножество $\{\alpha_1 a + \alpha_2 a^2 + \cdots + \alpha_n a^n \mid n \in N \ \& \ a_1, ..., a_n \in P\}$ является подалгеброй в A, содержащей элемент a. Следовательно,

$\langle a \rangle \subseteq \{\alpha_1 a + \alpha_2 a^2 + \cdots + \alpha_n a^n \mid n \in N \ \& \ \alpha_1, \ldots, \alpha_n \in P\}$.

Пусть A – алгебра над полем P. Обозначим множество всех ее подалгебр через $L(A)$ и определим на этом множестве две бинарные операции.

$$\forall B, C \in L(A) \quad B \wedge C = \{x \in A \mid x \in B \ \& \ x \in C\} = B \cap C,$$

$$\forall B, C \in L(A) \quad B \vee C = \langle B \cup C \rangle.$$

Из данных определений и предложения 1.2.1 следует, что $B \vee C$ - наименьшая из подалгебр алгебры A, содержащая подалгебры B и C.

1.3. Изоморфизмы алгебр.

Определение 4. Пусть A и A' – алгебры над полем P. *Изоморфизмом* алгебры A на алгебру A' назовем биективное отображение φ множества A на множество A', удовлетворяющее следующим условиям:

1. $\forall a, b \in A \quad \varphi(a + b) = \varphi(a) + \varphi(b)$;

2. $\forall a,b \in A \quad \varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$;

3. $\forall \alpha \in P \ \forall a \in A \quad \varphi(\alpha a) = \alpha(\varphi(a))$.

Замечание 1. Из условий 1) и 2) следует, что изоморфные алгебры являются изоморфными кольцами, а из условий 1) и 3) следует, что изоморфные алгебры являются изоморфными векторными пространствами. Поэтому изоморфизмы алгебр над полем обладают всеми свойствами изоморфизмов колец и векторных пространств.

Теорема 3. *Любая алгебра с единицей ранга* n *над полем* P *изоморфна некоторой подалгебре алгебры* $M_n(P)$.

Доказательство. Пусть A – алгебра ранга n над полем P. Для любого элемента a из A определим отображение $\varphi_a : A \to A$ следующим образом: $\forall x \in A \varphi_a(x) = xa$ и докажем, что φ_a – линейное отображение. Действительно,

1. $\forall x, y \in A \quad \varphi_a(x+y) = (x+y)a = xa + ya = \varphi_a(x) + \varphi_a(y)$;

2. $\forall x \in A \ \forall \alpha \in P \quad \varphi_a(\alpha x) = (\alpha x)a = \alpha(xa) = \alpha(\varphi_a(x))$.

Заметим, что $\varphi_{a+b} = \varphi_a + \varphi_b$; $\varphi_{ab} = \varphi_a \varphi_b$; $\varphi_{\alpha a} = \alpha \varphi_a$.

Зададим теперь отображение $\psi : A \to \Phi_n$ по следующему правилу: $\forall a \in A \psi(a) = \varphi_a$ и докажем, что ψ – инъективный гомоморфизм. Действительно, пусть $a,b \in A$ и $\psi(a) = \psi(b)$. Тогда $\varphi_a = \varphi_b$ то есть $\forall x \in A \varphi_a(x) = \varphi_b(x)$. Это значит, что $xa = xb$. Подставляя в это равенство единичный элемент e алгебры A, получим: $a = b$. Следовательно, ψ - инъективное отображение.

Пусть $a,b \in A, \alpha \in P$. Тогда

1) $\psi(a+b) = \varphi_{a+b} = \varphi_a + \varphi_b = \psi(a) + \psi(b)$;

2) $\psi(ab) = \varphi_{ab} = \varphi_a \varphi_b = \psi(a)\,\psi(b)$;

3) $\psi(\alpha a) = \varphi_{\alpha a} = \alpha \varphi_a = \alpha \psi(a)$.

Следовательно ψ - гомоморфизм. Пусть $\psi(A)$ - гомоморфный образ алгебры A. Тогда, во-первых, $\psi(A) \cong A$, а во-вторых, $\psi(A)$ - подалгебра

алгебры Φ_n. Учитывая теперь то, что $\Phi_n \cong M_n(P)$, получим то, что требовалось доказать.

Теорема 4. *Пусть* A *– произвольная алгебра ранга* n *над полем* P. *Тогда существует алгебра* A^* *ранга* $n+1$ *с единицей над полем* P, *содержащая подалгебру, изоморфную алгебре* A.

Доказательство. Рассмотрим множество $A^* = P \times A$ и определим на нем следующие операции:

$$\forall(\alpha,a),(\beta,b)\in A^* \quad (\alpha,a)+(\beta,b)=(\alpha+\beta,\ a+b); \qquad (1.1)$$

$$\forall(\alpha,a),(\beta,b)\in A^* \quad (\alpha,a)(\beta,b)=(\alpha\beta,\ \alpha b+\beta a+ab); \qquad (1.2)$$

$$\forall(\alpha,a)\in A^* \ \forall\beta\in P \quad \beta(\alpha,a)=(\beta\alpha,\ \beta a). \qquad (1.3)$$

Легко проверяется, что $(A^*,+)$ - абелева группа. Проверим выполнимость свойства дистрибутивности умножения относительно сложения:

$$\forall(\alpha,a),(\beta,b),(\gamma,c)\in A^* \quad (\alpha,a)\big((\beta,b)+(\gamma,c)\big)=(\alpha,a)(\beta+\gamma,\ b+c)=$$
$$=\big(\alpha(\beta+\gamma),\ \alpha(b+c)+(\beta+\gamma)a+a(b+c)\big)=(\alpha\beta+\alpha\gamma,\ \alpha b+\alpha c+\beta a+\gamma a+ab+ac)=$$
$$=(\alpha\beta,\ \alpha b+\beta a+ab)+(\alpha\gamma,\ \alpha c+\gamma a+ac)=(\alpha,a)(\beta,b)+(\alpha,a)(\gamma,c).$$

Следовательно, $(A^*,+,\cdot)$ - кольцо. Кроме того, не трудно проверить, что множество A^* относительно операций (1.1) и (1.3) образует векторное пространство.

Свойство 3) определения 1 также выполнимо в A^*:

$$\forall\gamma\in P \ \forall(\alpha,a),(\beta,b)\in A^* \quad \big(\gamma(\alpha,a)\big)(\beta,b)=(\gamma\alpha,\gamma a)(\beta,b)=(\gamma\alpha\beta,\ \gamma\alpha b+\beta\gamma a+(\gamma a)b)=$$
$$=\gamma(\alpha\beta,\ \alpha b+\beta a+ab)=\gamma(\alpha,a)(\beta,b)=(\alpha,a)\big(\gamma(\beta,b)\big).$$

Таким образом, A^* – алгебра над полем P. Очевидно, что элемент $(1,0)$ является единицей этой алгебры.

Определим размерность алгебры A^*. Пусть $e_1,e_2,...,e_n$ – базис алгебры A (то есть базис векторного пространства A над полем P). Тогда система векторов $(1,0),(0,e_1),(0,e_2),\ldots,(0,e_n)$ алгебры A^* линейно независима над полем P. Кроме того, $\forall(\alpha,a)\in A^* \ \exists\alpha,\alpha_1,\ldots,\alpha_n\in P \ \ (\alpha,a)=\alpha(1,0)+\alpha_1(0,e_1)+\cdots+\alpha_n(0,e_n)$. Следовательно, A^* - алгебра ранга $n+1$.

Рассмотрим теперь отображение $\psi : A \to A^*$, определенное следующим образом: $\forall a \in A \;\; \psi(a) = (0, a)$. Легко видеть, что ψ – инъективное отображение.

Пусть $a, b \in A, \; \alpha \in P.$ Тогда

1) $\psi(a+b) = (0, a+b) = (0, a) + (0, b) = \psi(a) + \psi(b)$;

2) $\psi(ab) = (0, ab) = (0, a)(0, b) = \psi(a)\psi(b)$;

3) $\psi(\alpha a) = (0, \alpha a) = \alpha(0, a) = \alpha\psi(a)$;

Следовательно, ψ – гомоморфизм и потому алгебра A изоморфна подалгебре $\psi(A)$ алгебры A^*.

Из теорем 3 и 4 вытекает следующая теорема.

Теорема 5. *Любая алгебра ранга n над полем P изоморфна некоторой подалгебре алгебры $M_{n+1}(P)$.*

1.4. Решетки подалгебр.

Определение 5. *Решеткой* (или *структурой*) называется непустое множество L с определенными на нем двумя бинарными операциями $\vee$ и $\wedge$, удовлетворяющими следующим условиям:

1) $\forall a \in L \;\; a \vee a = a, \;\; a \wedge a = a$ идемпотентность;

2) $\forall a, b \in L \;\; a \vee b = b \vee a, \;\; a \wedge b = b \wedge a$ коммутативность;

3) $\forall a, b, c \in L \;\; a \vee (b \vee c) = (a \vee b) \vee c, \;\; a \wedge (b \wedge c) = (a \wedge b) \wedge c$ ассоциативность;

4) $\forall a, b \in L \;\; a \vee (a \wedge b) = a, \;\; a \wedge (a \vee b) = a$ свойства поглощения.

Операции $\vee$ и $\wedge$, используемые в определении решетки, будем называть *решеточным объединением* и *решеточным пересечением* соответственно.

Пусть A – алгебра над полем P. Обозначим множество всех ее подалгебр через $L(A)$ и определим на этом множестве две бинарные операции.

$\forall B, C \in L(A) \; B \wedge C = \{x \in A \mid x \in B \;\&\; x \in C\}$,

$\forall B, C \in L(A) \; B \vee C = \langle B \cup C \rangle$.

Из данных определений следует, что $B \wedge C = B \cap C$ и согласно предложению $B \wedge C$ - подалгебра алгебры A; $B \vee C$ - наименьшая из подалгебр алгебры A, содержащая подалгебры B и C.

Теорема 6. *Пусть A – алгебра над полем P. Тогда $(L(A), \wedge, \vee)$ - решетка.*

Доказательство. Проверим выполнимость четырех аксиом решетки для $(L(A), \wedge, \vee)$.

1) Идемпотентность. $\forall B \in L(A)\ B \wedge B = B \cap B = B;\ B \vee B = \langle B \cup B \rangle = \langle B \rangle = B.$

2) Коммутативность. $\forall B, C \in L(A)\ B \wedge C = B \cap C = C \cap B = C \wedge B;$

$$B \vee C = \langle B \cup C \rangle = \langle C \cup B \rangle = C \vee B.$$

3) Ассоциативность.

$$\forall B, C, D \in L(A)\ B \wedge (C \wedge D) = B \cap (C \cap D) = (B \cap C) \cap D = (B \wedge C) \wedge D;$$

$$B \vee (C \vee D) = B \vee \langle C \cup D \rangle = \langle B \cup \langle C \cup D \rangle \rangle = \langle B \cup (C \cup D) \rangle = \langle (B \cup C) \cup D \rangle =$$

$$\langle \langle B \cup C \rangle \cup D \rangle = \langle B \cup C \rangle \vee D = (B \vee C) \vee D.$$

4) Поглощение.

$$\forall B, C \in L(A)\ B \wedge (B \vee C) = B \cap \langle B \cup C \rangle = B;$$

$$B \vee (B \wedge C) = B \vee (B \cap C) = B.$$

1.5. Алгебраические элементы колец

Любая алгебра над полем является кольцом с операциями сложения (+) и умножения (·). Обозначим: $A = (A, +, \cdot),\ P = GF(2) = \{0, 1\}$.

Определение 6. Элемент e кольца K называется *идемпотентным* элементом, если $e^2 = e$.

Пример 3. В кольце целых чисел всего два идемпотентных элемента: 0 и 1.

Пример 4. В кольце квадратных матриц порядка 2 следующая матрица является идемпотентной $m = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, так как $m^2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$.

Определение 7. Элемент r кольца K называется *нильпотентным*, если $(\exists n \in \mathrm{N})\, r^n = 0$. Наименьшее число n с таким свойством называется индексом нильпотентности элемента n ($ind\ r = n$).

Пример нильпотентных элементов и их индексов.

1) $0^2 = 0,\ ind\ 0 = 1;$

2) $r = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} r^2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = 0, ind\ r = 2;$

3) $m = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} m^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} m^3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, ind\ m = 3;$

Определение 8. Элемент a кольца K называется *алгебраическим*, если существует многочлен положительной степени $f(x)$ с целым коэффициентом, то есть $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x$ такой, что $f(a) = 0$.

1.6. Пирсовские разложения колец

Пусть K – коммутативное кольцо, e – ненулевой идемпотентный элемент.

Определим два множества:

$$eK = \{ex | x \in K\} \neq \emptyset$$

$$(1-e)K = \{x - ex | x \in K\} \neq \emptyset$$

Докажем, что eK и $(1-e)K$ – подкольца в K.

Признак подкольца:

$$\forall a, b \in S\ \big((a-b) \in S\big)$$

$$\forall a, b \in S\ (ab \in S)$$

Пусть $S = eK, a = ex_1, b = ex_2$.

$$a - b = e(x_1 - x_2) \in eK$$

$$ab = ex_1 ex_2 = e^2 x_1 x_2 = e(ex_1 x_2) \in eK$$

Пусть $S = (1-e)K, a = (1-e)x_1, b = (1-e)x_2$.

$$a - b = x_1 - x_2 - e(x_1 - x_2) = (1-e)(x_1 - x_2) \in (1-e)K$$

$$ab = (1-e)x_1(1-e)x_2 = (1-e)((1-e)x_1 x_2) \in (1-e)K$$

$$eK + (1-e)K = K - \text{докажем это:}$$

Пусть $x \in K$. Тогда $x = ex + (1-e)x = ex + x - ex = x$.

Значит $K \subseteq eK + (1-e)K$. Так как $eK, (1-e)K \subseteq K$, то $eK + (1-e)K = \{ex + (1-e)y = ex + y - ey | x, y \in K\} \subseteq K$.

Убедимся в том, что $eK \cap (1-e)K = \{0\}$.

Пусть $a \in eK \cap (1-e)K$. Тогда $\exists\, x, y \in K$ такие, что

$$a = ex = (1-e)y$$

$$ea = e(ex) = e(1-e)y = e(y - ey) = ey - ey = 0 \Rightarrow a = 0$$

Обозначим $eK + (1-e)K = eK \oplus (1-e)K$ – прямая сумма двух подколец.

Таким образом: $K = eK \oplus (1-e)K$ является пирсовским разложением кольца K по идемпотенту e. Легко видеть, что $\forall a \in eK\ ea = a,$ то есть e – единичный элемент в подкольце eK. Аналогично: $\forall c \in (1-e)K\ ec = 0$. Значит, если $x \in eK$, а $y \in (1-e)K$, то $e(x+y) = x \Rightarrow xy = 0$.

Пусть K – некоммутативное кольцо, e – идемпотентный элемент. Тогда, если e – не единица, то имеет место двустороннее пирсовское разложение:

$$K = eKe \oplus eK(1-e) \oplus (1-e)Ke \oplus (1-e)K(1-e).$$

Глава 2. Система компьютерной алгебры GAP

2.1. Общая характеристика пакета GAP

GAP (Groups, Algorithms and Programming) является системой компьютерной алгебры, задуманной как инструмент вычислительной теории групп, и впоследствии распространившейся на смежные разделы алгебры. Первоначально GAP разрабатывался в г.Аахен, Германия (Lehrstuhl D fur Mathematik, RWTH). В настоящее время центр разработки GAP и технической поддержки его пользователей находится в Шотландии (School of Mathematical and Computational Sciences, University of St.-Andrews).

Основные особенности GAP:

- язык программирования, внешне напоминающий Паскаль;
- стандартные типы основных алгебраических объектов: групп (подстановок, абстрактных, матричных), колец, полей;
- удобные типы переменных, в т.ч. оперативно изменяемые списки и записи;
- более 4 тысяч библиотечных функций;
- обширная библиотека данных, включая практически все группы, порядок которых не превосходит 1000;
- прикладные программы, поставляемые вместе с GAP, охватывают такие разделы алгебры, как комбинаторная теория групп, конечные простые группы, теория представлений групп, теория графов, в т.ч. их группы автоморфизмов, теория кодирования, кристаллографические группы, группы Галуа и многое другое;
- подробное и удобное описание (около 1600 стр.) в формате «гипертекст»;
- бесплатное получение по сети Internet вместе с исходными текстами, являющимися незаменимым наглядным пособием для освоения GAP;
- работа в операционных системах DOS, Windows, Unix, Linux, MacOS;
- работа с процессором типа 386 и выше с ОЗУ от 8 Mb;

– занимаемое место на диске – от 10 до 100 Mb в зависимости от объема инсталляции;
– способность работать с ОЗУ до 128 Mb и файлом подкачки до 128 Mb;

Запуск GAP и выход из системы

Запуск GAP в MS-DOS осуществляется с помощью командного файла gap.bat, который должен находиться в каталоге, указанном в команде PATH в файле autoexec.bat. Если во время работы с GAP необходимо чтение программ (файлов с расширением "g"), перед запуском GAP рекомендуется перейти в содержащий эти программы каталог (здесь и далее предполагается, что читатель уже владеет необходимыми навыками работы с ПЭВМ).

При успешном запуске GAP на экране появится эмблема GAP и приглашение системы, которое имеет следующий вид:

gap>

Для выхода из системы применяется команда quit; (заметим, что любая команда завершается точкой с запятой).

Примечание. Для дублирования введенных команд и выводимых на экран результатов в текстовом файле используется команда LogTo("filename.log");. Ведение файла протокола может быть остановлено командой LogTo(); (например, чтобы просмотреть его содержимое в другом окне Windows, не прерывая сеанса работы с GAP).

Примеры простейших вычислений

Вы можете использовать GAP в качестве обыкновенного калькулятора:

Пример 1:

```
gap> (9 - 7) * (5 + 6);
22
gap>
```

Пример 2:

```
gap> (9 - 7) * (5 + 6)
> ; # знак ">" - промежуточное приглашение GAP
22
```

gap>

2.2. Общие команды пакета.

Символы и категории слов в GAP

GAP воспринимает следующие символы: цифры, буквы (верхний и нижний регистры), пробел, символы табуляции и новой строки, а также специальные символы:

```
"   '   (   )   *   +   ,   -
.   /   :   ;   <   =   >   ~
{   \   ]   ^   _   {   }   #
```

Составленные из символов слова относятся к следующим категориям:

- ключевые слова (зарезервированные последовательности букв нижнего регистра)
- идентификаторы (последовательности цифр и букв, содержащая не менее одной буквы и не являющаяся ключевым словом)
- строки (последовательности произвольных символов, заключенная в двойные кавычки)
- целые числа (последовательности цифр)

операторы и ограничители в соответствии со следующим списком:

```
+   -   *   /   ^   ~
=   <>  <   <=  >   >=
:=  .   . .  ->  ,   ;
[   ]   {   }   (   )
```

Следует заметить, что пробелы могут быть использованы для повышения удобочитаемости текста, так как любая последовательность пробелов воспринимается GAP как один пробел. Таким образом, команда

```
if i<0 then a:=-i;else a:=i;fi;
```

может быть записана следующим образом:

```
if i < 0 then    # если i отрицательное
  a := -i;
```

```
else        # иначе
  a := i;
fi;
```

Ключевые слова

Ключевыми словами GAP являются следующие слова:

```
and   do    elif   else   end   fi
for   function   if   in   local   mod
not   od   or   repeat return then
until   while quit
```

Идентификаторы

Идентификаторы состоят из букв, цифр, символов «_», и должны содержать не менее одной буквы или символа «_». При этом регистр является существенным. Примеры идентификаторов:

```
a       foo    LongIdentifier
hello   Hello  HELLO
x100    100x   _100
underscores_case   MixedCase
```

Выражения

Примерами выражений являются: переменные, обращения к функциям, целые числа, перестановки, строки, функции, списки, записи. С помощью операторов из них могут быть составлены более сложные выражения. Операторы разбиты на три класса:

– операторы сравнения: = <> <= in
– арифметические операторы: + - * / mod ^
– логические операторы: not and or

Пример 3:

```
gap>2*2;; #два знака ";" подавляют вывод на экран
gap>2*2+9=Fibonacci(7) and Fibonacci(13) in Prime;
true
```

Следует различать глобальные и локальные переменные, различия которых можно видеть из следующего примера:

Пример 4:

```
g := 0;      # глобальная переменная g
x := function ( a, b, c )
  local  y;
  g := c;  # c - аргумент функции x
  y := function ( y )
    local d, e, f;
    d := y; # y - аргумент функции y
    e := b; # b - аргумент функции x
    f := g; # g - глобальная переменная g
    return d + e + f;
  end;
  return y(a); # y-локальная переменная функции x
end;
```

Обращения к функциям

Формат:

```
   function-var()
   function-var( arg-expr {, arg-expr} )
```

Пример:

```
  gap> Fibonacci( 11 );
  # обращение к функции "Fibonacci" с аргументом 11
  89
  gap> G.operations.RightCosets(G,Intersection(U,V));;
  #обращение к функции "G.operations.RightCosets",
  #в котором второй аргумент определяется
  #обращением к другой функции
```

Сравнение выражений

Формат:

left-expr = right-expr

left-expr <> right-expr

Примечание: любые объекты сравнимы между собой. Объекты различных типов всегда различны, т.е. = приведет к false, и <> — к true. Кроме того, для них определено отношение «меньше».

Операторы сравнения имеют больший приоритет по сравнению с логическими операторами, но меньший по сравнению с арифметическими. Например, a*b = c and d интерпретируется как ((a*b)=c) and d). Еще один пример (сравнение, левая часть которого является выражением):

gap> 2 * 2 + 9 = Fibonacci(7);

true

Арифметические операторы

Формат:

+ right-expr

- right-expr

eft-expr + right-expr

left-expr - right-expr

left-expr * right-expr

eft-expr / right-expr

left-expr mod right-expr

left-expr ^ right-expr

Значение, как правило, зависит от типа операндов. Mod определен только для целых и рациональных чисел. Для элемента группы ^ означает возведение в степень, если правый операнд — целое число, а если он — также элемент группы, то сопряжение с его помощью. Приоритет операторов (по убыванию):

1. ^
2. унарные + и -
3. *, /, mod
4. + и -

Пример 5: -2 ^ -2 * 3 + 1 означает (-(2 ^ (-2)) * 3) + 1 .

Арифметические операторы имеют наивысший приоритет по сравнению с операторами сравнения и логическими операторами.

Присваивания

Командами в GAP называются: присваивания, вызовы процедур, структуры if, while, repeat, for, а также команда return. Все команды заканчиваются знаком « ; ».

Присваивания имеют формат

```
  var := expr;
```

Пример:

```
gap> S6 := rec( size := 720 );; S6;
rec(
 size := 720 )
gap> S6.generators := [ (1,2), (1,2,3,4,5) ];; S6;
rec(
 size := 720,
 generators := [ (1,2), (1,2,3,4,5) ] )
gap> S6.generators[2] := (1,2,3,4,5,6);; S6;
rec(
 size := 720,
 generators := [ (1,2), (1,2,3,4,5,6) ] )
```

Вызов процедуры

Формат:

```
   procedure-var();
   procedure-var( arg-expr {, arg-expr} );
```

Различие между процедурами и функциями введено для удобства, GAP же их не различает. Функция возвращает значение, но не производит побочных эффектов. Процедура не возвращает никакого значения, но производит какое-либо действие (например, процедуры Print, Append, Sort).

Команда IF

Формат:

```
if bool-expr1 then statements1
{ elif bool-expr2 then statements2 }
[ else statements3 ]
fi;
```

При этом частей elif может быть произвольное количество или ни одной. Часть else также может отсутствовать.

Пример 1: в командах

```
if expr1 then
  if expr2 then stats1
  else stats2 fi;
fi;
```

else относится ко второму if, тогда как в командах

```
if expr1 then
  if expr2 then stats1 fi;
else stats2
fi;
```

else относится к первому if.

Пример 2:

```
gap> i := 10;;
gap> if 0 < i then
>    s := 1;
>  elif i < 0 then
>    s := -1;
>  else
>    s := 0;
>  fi;
gap> s;
1    # знак i
```

Цикл WHILE

Формат:

while bool-expr do statements od;

Последовательность команд statements выполняется, пока истинно условие bool-expr. При этом сначала проверяется условие, а затем, если оно истинно, выполняются команды. Если уже при первом обращении условие ложно, то последовательность команд statements не выполнится ни разу.

Пример:

```
gap> i := 0;; s := 0;;
gap> while s <= 200 do
>    i := i + 1; s := s + i^2;
>  od;
gap> s;
204
```

Цикл REPEAT

Формат:

repeat statements until bool-expr;

Последовательность команд statements выполняется, пока истинно условие bool-expr. При этом сначала выполняются команды, а затем проверяется условие. Таким образом, при любом начальном значении условия набор команд statements выполнится, по крайней мере, один раз.

Пример (вычисление наименьшей суммы квадратов первых n последовательных натуральных чисел, превышающей 200):

```
gap> i := 0;; s := 0;;
gap> repeat
>    i := i + 1; s := s + i^2;
>  until s > 200;
gap> s;
204
```

Цикл FOR

Формат:

for simple-var in list-expr do statements od;

При этом последовательность команд statements выполняется для каждого элемента из списка list-expr. Цикл for эквивалентен циклу while:

```
loop-list := list;
loop-index:= 1;
while loop-index <= Length(loop-list) do
 variable := loop-list[loop-index];
 ...
statements
 ...
 loop-index := loop-index+1;
od;
```

Список list часто является последовательностью. Команда

for variable in [from..to] do statements od;

соответствует распространенной в других языках команде

for variable from from to do statements od;

Пример:

```
gap> s := 0;;
gap> for i  in [1..100] do
>    s := s + i;
> od;
gap> s;
5050
```

В следующем примере изменение списка приводит к выполнению команд для новых его элементов:

```
gap> l := [ 1, 2, 3, 4, 5, 6 ];;
gap> for i in l  do
>    Print( i, " " );
>    if i mod 2 = 0 then Add( l, 3*i/2 ); fi;
> od; Print( "\n" );
```

```
1 2 3 4 5 6 3 6 9 9
gap> l;
[ 1, 2, 3, 4, 5, 6, 3, 6, 9, 9 ]
```

А в следующем — не приводит:

```
gap> l := [ 1, 2, 3, 4, 5, 6 ];;
gap> for i  in l do
>    Print( i, " " );
>    l := [];
> od; Print( "\n" );
1 2 3 4 5 6
gap> l;
[ ]
```

Функции

Формат:

```
   function ( [ arg-ident {, arg-ident} ] )
         [ local loc-ident {, loc-ident} ; ]
         statements
         end
```

Пример функции, которая определяет n-е число Фибоначчи:

```
gap> fib := function ( n )
>     local f1, f2, f3, i;
>     f1 := 1; f2 := 1;
>     for i in [3..n] do
>       f3 := f1 + f2; f1 := f2; f2 := f3;
>     od;
>     return f2;
>   end;;
gap> List( [1..10], fib );
[ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ]
```

Ту же функцию можно определить рекурсивно:

```
gap> fib := function ( n )
>    if n < 3 then
>      return 1;
>    else
>      return fib(n-1) + fib(n-2);
>    fi;
>  end;;
gap> List( [1..10], fib );
[ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ]
```

Заметим, что рекурсивная версия требует 2 * fib(n)-1 шагов для вычисления fib(n), тогда как итеративная требует только n-2 шага. Обе, однако, не являются оптимальными, так как библиотечная функция Fibonacci требует порядка Log(n) шагов.

Запись arg-ident -> expr является краткой записью для функции

function (arg-ident) return expr; end.

Здесь arg-ident — один идентификатор, т.е. таким образом нельзя задать функцию от нескольких переменных.

Пример типичного использования такой записи:

```
gap> Sum( List( [1..100], x -> x^2 ) );
338350
```

Команда RETURN

Формат:

return;

return expr;

Первая форма прерывает выполнение внутренней (при вызове одной функции из другой) функции и передает управление вызывающей функции, не возвращая при этом никакого значения. Вторая, кроме того, возвращает значение выражения expr.

2.3. Команды для вычислений в алгебрах

Алгебры в GAPe

Пусть F — поле и A — алгебра над F (кратко: A — F -алгебра). Все алгебры в GAPe ассоциативны, то есть операция умножения в них ассоциативна. Любая алгебра всегда содержит нулевой элемент, который может быть получен, вычитанием произвольного элемента из самого себя. Элементы поля F не рассматриваются как элементы A. Практическим обоснованием (очевидно и математическим тоже) для этого служит то, что даже если единичная матрица содержится в матричной алгебре A, все равно не возможно записать 1 + a для суммы единичной матрицы и элемента a алгебры A, так как независимо от алгебры A в GAPe уже определено это значение как прибавление 1 ко всем позициям матрицы a. Вместо этого необходимо писать

One(A) + a или

aˆ0 + a.

Алгебры и унитальные алгебры

Не все алгебры содержат (левый и правый) мультипликативный нейтральный единичный элемент, но если алгебра содержит такой единичный элемент, то он единственен. Если алгебра A содержит мультипликативный нейтральный элемент, то в общем случае он не может быть получен из произвольного элемента a алгебры A как частное a/a или как a0, так как эти операции могут быть не определены для алгебры A. Более точно, может быть возможно инвертировать a или возвести его в нулевую степень, но A может быть не замкнуто относительно этих операций. Например, если a — квадратная матрица в GAPe, тогда мы можем считать, что a0 является единичной матрицей того же самого размера и над тем же самым полем, что и a. С другой стороны, алгебра может иметь мультипликативный нейтральный элемент, который не равен нулевой степени элементов. Во многих случаях, однако, нулевая степень элементов алгебры правильно определена как элемент алгебры. Это справедливо,

например, для всех тех матричных алгебр, чьи порождающие элементы являются порождающими элементами конечной группы.

Для практических целей полезно различить общие алгебры и унитальные алгебры.

Унитальная алгебра в GAPe — алгебра U , которая содержит нулевую степень элементов, и в которой выполнимы все действия с этой степенью. Не унитальная алгебра A может не содержать нулевые степени элементов или, наоборот содержать их, но тогда в ней невыполнимо какое-нибудь действие с этой степенью. Итак, возможно рассматривать A как унитальную алгебру, используя AsUnitalAlgebra (A), и, конечно, всегда возможно рассматривать унитальную алгебру как алгебру, используя AsAlgebra (U). Алгебра A может иметь унитальные подалгебры, и, конечно, алгебра U может иметь подалгебры, которые не являются унитальными. Образы унитальных алгебр при гомоморфизмах являются или унитальными, или тривиальными, так как образ единицы при гомоморфизме есть единица. Следующий пример показывает главные различия между алгебрами и унитальными алгебрами.

```
gap> a:= [ [ 1, 0 ], [ 0, 0 ] ];;
gap> alg1:= Algebra( Rationals, [ a ] );
Algebra( Rationals, [ [ [ 1, 0 ], [ 0, 0 ] ] ] )
gap> id:= a^0;
[ [ 1, 0 ], [ 0, 1 ] ]
gap> id in alg1;
false
gap> alg2:= UnitalAlgebra( Rationals, [ a ] ); UnitalAlgebra( Rationals, [ [ [
1, 0 ], [ 0, 0 ] ] ] ) gap> id in alg2;
true
gap> alg3:= AsAlgebra( alg2 );
Algebra( Rationals, [ [ [ 1, 0 ], [ 0, 0 ] ], [ [ 1, 0 ], [ 0, 1 ] ]
] )
gap> alg3 = alg2;
```

true

gap> AsUnitalAlgebra(alg1); Error, <D> is not unital

Мы видим, что, если мы хотим, чтобы единичная матрица содержалась в алгебре, о которой неизвестно является ли она унитальной, нам необходимо прибавить ее к порождающим элементам.

Родительские алгебры и подалгебры

GAP различиет алгебры и подалгебры алгебр.

Каждая подалгебра принадлежит уникальной основной алгебре, которую называют родителем подалгебры. Родительская алгебра — собственный родитель. Родительские алгебры конструируются при помощи операторов Algebra и UnitalAlgebra, подалгебры конструируются при помощи операторов Subalgebra и UnitalSubalgebra. Родитель первого аргумента оператора Subalgebra будет родителем созданной подалгебры. Алгебраические действия, совершаемые более, чем с одной алгеброй, предполагают, что аргументы имеют общего родителя. Возьмем, например, Centralizer. В этом случае должно быть два аргумента: алгебра A и алгебра B, где A родительская алгебра и B — подалгебра этой родительской алгебры, или A и B — подалгебры общей родительской алгебры P. В этих случаях Centralizer выдает централизатор B в A, который представлен как подалгебра общей родительской алгебры алгебр A и B. Заметим, что подалгебра родительской алгебры не должна быть собственной подалгеброй. Исключением этому правилу является теоретико-множественная функция Intersection, которая позволяет рассматривать пересечения алгебры с различными родительскими алгебрами. Всякий раз, когда имеется две подалгебры, которые имеют различные родительские алгебры, но имеют и общую супералгебру A, можно использовать AsSubalgebra или AsUnitalSubalgebra для того, чтобы создать новые подалгебры, которые имеют общую родительскую алгебру A.

Algebra

Algebra(U) выдает родительскую алгебру A, которая изоморфна родительской алгебре или подалгебре U.

Algebra(F, gens)

Algebra(F,gens, zero) выдает родительскую алгебру над полем F и порожденную элементами алгебры в списке gens. Нулевой элемент этой алгебры может быть введен как zero; это необходимо всякий раз, когда gens пусто.

gap> a:= [[1]];;

gap> alg:= Algebra(Rationals, [a]); Algebra(Rationals, [[[1]]])

gap> alg.name:= "alg";;

gap> sub:= Subalgebra(alg, []); Subalgebra(alg, [ ])

gap> Algebra(sub);

Algebra(Rationals, [[[0]]]) gap> Algebra(Rationals, [], 0*a); Algebra(Rationals, [[[0]]])

Алгебры, получаемые с помощью Algebra, не унитальны. Для построения унитальных алгебр используйте UnitalAlgebra.

Subalgebra

Subalgebra(A, gens) выдает подалгебру алгебры A, порожденную элементами в списке gens.

gap> a:= [[1, 0], [0, 0]];;

gap> b:= [[0, 0], [0, 1]] ;;

gap> alg:= Algebra(Rationals, [a, b]);;

gap> alg.name:= "alg";;

gap> s:= Subalgebra(alg, [a]);

Subalgebra(alg, [[[1, 0], [0, 0]]])

gap> s = alg;

false

gap> s:= UnitalSubalgebra(alg, [a]); UnitalSubalgebra(alg, [[[1, 0], [0, 0]]]) gap> s = alg;

true

Теоретико-множественные функции для алгебр

Как уже упомянуто во введении главы, алгебры являются областями. Таким образом, все теоретико-множественные функции, например, Intersection и Size могут быть применены к алгебрам. Все теоретико-множественные функции, не упомянутые здесь, не трактуются специально для алгебр.

Elements(A) вычисляет элементы алгебры A с использованием алгоритма Dimino. Заданная по умолчанию для алгебр функция вычисляет базис линейного пространства в то же самое время.

Intersection(A, H) выдает пересечение A и H в виде множества элементов или как алгебраическую запись (запись алгебры).

IsSubset (A, H)

Если A и H — алгебры, то IsSubset проверяет являются ли генераторы H элементами A. Другой способ состоит в применении DomainOps.IsSubset.

Random(A) выдает произвольный элемент алгебры A. Это требует вычисления базиса линейного пространства.

Проверка свойств алгебр

С помощью GAPa могут быть проверены следующие свойства алгебр.

IsAbelian(A) выдается true если алгебра A абелева и false в противном случае. Алгебра A называется абелевой, если и только, если для любых a, b $\in$ A $a * b = b * a$.

IsCentral(A, U) выдается true если алгебра A централизует алгебру U и false в противном случае. Алгебра A централизует алгебру U , если и только, если для всякого $a \in A$ и для всякого $u \in U$ $a * u = u * a$. Заметьте, что U не обязана быть подалгеброй A, но они должны иметь общую родительскую алгебру.

IsFinite(A) выдается true если алгебра A конечна, и false в противном случае.

IsTrivial(A) выдается true если алгебра A состоит только из нулевого элемента, и false в противном случае. Если A — унитальная алгебра, то, конечно, она никогда не тривиальна.

Все критерии ожидают родительскую алгебру или подалгебру и выдают true, если алгебра имеет свойство и false в противном случае. Некоторые функции не могут выполниться, если данная алгебра имеет бесконечное множество элементов. В таких случаях может быть напечатано предупреждение.

gap> IsAbelian(FreeAlgebra(GF(2), 2));

false

gap> a:= UnitalAlgebra(Rationals, [[[1, 0], [0, 0]]]); UnitalAlgebra(Rationals, [[[1, 0], [0, 0]]])

gap> a.name:= "a";;

gap> s1:= Subalgebra(a, [One(a)]); Subalgebra(a, [[[1, 0], [0, 1]]]) gap> IsCentral(a, s1); IsFinite(s1); true

false

gap> s2:= Subalgebra(a, []); Subalgebra(a, [ ])

gap> IsFinite(s2); IsTrivial(s2);

true true

Функции линейного пространства для алгебр

Конечно-мерная F-алгебра A всегда есть конечно-мерное векторное пространство над F. Таким образом, в GAPe, алгебра — линейное пространство, и функции линейного пространства типа Base и Dimension применимы к алгебрам.

gap> a:= UnitalAlgebra(Rationals, [[[1, 0], [0, 0]]]); UnitalAlgebra(Rationals, [[[1, 0], [0, 0]]])

gap> Dimension(a);

2

gap> Base(a);

[[[1, 0], [0, 1]], [[0, 0], [0, 1]]]

Структура линейного пространства используется также теоретико-множественными функциями.

Алгебраические функции для алгебр

Функции, описанные в этом разделе, вычисляют некоторые подалгебры данной алгебры, например, Centre вычисляет центр алгебры. Некоторые функции не могут завершиться, если данная алгебра имеет бесконечное множество элементов, в то время как другие функции могут сообщить об ошибке в таких случаях.

В GAPe каждая алгебра является или родительской алгеброй или подалгеброй единственной родительской алгебры. Если Вы вычисляете центр C алгебры U с родительской алгеброй A, то C — подалгебра U , но ее родительская алгебра есть A.

Centralizer(A, x)

Centralizer(A, U) выдают централизатор элемента x в A, где x должен быть элементом родительской алгебры A, соответственно централизатор алгебры U в A, где обе алгебры должны иметь общего родителя.

Централизатор элемента x в A определен как множество C элементов c из A, таких, что c и x коммутируют.

Централизатор алгебры U в A определен как множество C элементов c из A, та- ких, что c коммутирует с каждым элементом U .

```
gap> a:= MatAlgebra( GF(2), 2 );;
gap> a.name:= "a";;
gap> m:= [ [ 1, 1 ], [ 0, 1 ] ] * Z(2);;
gap> Centralizer( a, m );
UnitalSubalgebra( a, [ [ [ Z(2)^0, 0*Z(2) ], [ 0*Z(2), Z(2)^0 ] ], [ [ 0*Z(2), Z(2)^0 ], [ 0*Z(2), 0*Z(2) ] ] ] )
```

Centre(A)

выдает центр A (то есть централизатор A в A).

```
gap> c:= Centre( a );
UnitalSubalgebra( a, [ [ [ Z(2)^0, 0*Z(2) ], [ 0*Z(2), Z(2)^0 ] ]] )
```

Closure(U , a) Closure(U , S)

Пусть U — алгебра с родительской алгеброй A и пусть a — элемент A. Тогда Closure выдает замыкание C алгебры U и элемента a как подалгебру алгебры A. Замыкание C алгебры U и элемента a — подалгебра, порожденная U и a.

Пусть U и S две алгебры с общей родительской алгеброй A. Тогда Closure выдает подалгебру A, порожденную U и S.

gap> Closure(c, m);

UnitalSubalgebra(a, [[[Z(2)^0, 0*Z(2)], [0*Z(2), Z(2)^0]], [[Z(2)^0, Z(2)^0], [0*Z(2), Z(2)^0]]])

TrivialSubalgebra

TrivialSubalgebra(U)

Пусть U — алгебра с родительской алгеброй A. Тогда TrivialSubalgebra выдает три- виальную подалгебру T алгебры U , как подалгебру алгебры A.

gap> a:= MatAlgebra(GF(2), 2);;

gap> a.name:= "a";;

gap> TrivialSubalgebra(a); Subalgebra(a, [ ])

Элементы алгебры

Этот раздел описывает операции и функции, доступные для элементов алгебры. Заметьте, что элементы алгебры могут существовать независимо от алгебры, например, Вы можете записывать две матрицы и вычислять их сумму и произведение без когда-либо определения алгебры, которая содержит их.

Сравнения элементов алгебры

g = h выдает true, если элементы алгебры g и h равны и false в противном случае.

g <> h выдает true, если элементы алгебры g и h не равны и false в противном случае.

g < h

g <= h g >= h g > h

Операторы <, <=, >= и > выдают true, если элемент g — строго меньше, меньше или равен, больше или равен и строго больше, чем элемент h. Общего упорядочения всех элементов алгебры может не быть, но g и h должны лежать в одной родительской алгебре.

Арифметические операции для элементов алгебры

a ∗ b

a + b a − b

Операторы * , + и - вычисляют произведение, сумму и разность двух элементов алгебры a и b. Операнды должны конечно лежать в общей родительской алгебре, в противном случае выдается сообщение об ошибке.

a/c выдает частное элемента a и ненулевого элемента c основного поля алгебры (поля частных алгебры?).

a ^i выдает i -ию степень элемента a для положительного целого числа i. Если число i — нуль или отрицательно, то возможно результат не определен, или не содержится в алгебре, порожденной элементом a.

list + a a + list

list ∗ a a ∗ list

В этой форме операторы + и * выдают новый список, где каждая запись — сумма и соответственно произведение элемента a и соответствующего элемента списка. Конечно сложение и соответственно умножение должно быть определено между a и каждым элементом списка.

Глава 3. Моногенные подалгебры матричной алгебры $M_3(GF(2))$

Теорема 1. *В алгебре $M_3(GF(2))$ содержится всего 374 различных моногенных подалгебр, из которых*

1 подалгебра порядка 1 (нулевая подалгебра);

78 подалгебр порядка 2;

238 подалгебр порядка 4;

57 подалгебр порядка 8.

Доказательство. Согласно теореме Гамильтона – Кэли [2, с. 87] каждая квадратная матрица является корнем своего характеристического многочлена и потому порядок любой моногенной подалгебры в $M_3(GF(2))$ не может превосходить восьми. Дальнейшее доказательство проведено с помощью следующей программы:

Таблица № 1

Программа построения моногенных подалгебр		
№	**Текст программы**	**Коментарии**
1	`Mon:=[];`	Создание каталога для размещения моногенных подалгебр
2	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
3	`El:=Elements(A);`	Создание массива элементов алгебры A
4	`for i in [1..512] do`	Начало цикла построения моногенных подалгебр
5	`B:=Subalgebra(A,[El[i]]);`	Построение моногенной подалгебры
6	`el:=Elements(B);`	Создание ее элементов
7	`AddSet(Mon,el);`	Накопление в массиве Mon только новых подалгебр
8	`od;`	Конец цикла построения моногенных подалгебр
9	`m:=Size(Mon);`	Вычисления числа моногенных подалгебр
10	`N:=[];`	Создание массива для записей количеств подалгебр по всем возможным порядкам: 1, 2, 4, 8 (остальных порядков не может быть)
11	`for i in [1..3] do`	Начало разбивки массива N
12	`N[i]:=0;`	Обнуление начальных значений
13	`od;`	Конец разбивки массива N
14	`for i in [1..m] do`	Начало подсчета количества подалгебр
15	`s:=Size(Mon[i]);`	Вычисление числа элементов моногенной

		подалгебры из массива Mon
16	`if s=1 then N[1]:=N[1]+1;fi;`	Подсчет количества моногенных подалгебр порядка 1
17	`if s=2 then N[2]:=N[2]+1;fi;`	Подсчет количества моногенных подалгебр порядка 2
18	`if s=4 then N[3]:=N[3]+1;fi;`	Подсчет количества моногенных подалгебр порядка 4
19	`if s=8 then N[4]:=N[4]+1;fi;`	Подсчет количества моногенных подалгебр порядка 8
20	`od;`	Конец подсчета количеств подалгебр
21	`Print("N = ",N,"\n");`	Печать результата

Результат работы программы выглядит следующим образом: N = [1, 78, 238, 57]. Это доказывает теорему.

3.1. Моногенные подалгебры порядка 2

Теорема 2. *В алгебре $M_3(GF(2))$ содержится всего 78 различных моногенных подалгебр порядка 2, из которых*

21 подалгебра порождена нильпотентной матрицей;

57 подалгебр порождены идемпотентными матрицами.

Номера матриц, порождающих подалгебры порядка 2, приведены в таблице № 2.

Таблица № 2

Номера матриц, порождающих моногенные подалгебры второго порядка	
Номера 2-нильпотентных матриц	Номера идемпотентных матриц
3, 5, 7, 9, 28, 33, 37, 41, 64, 65, 73, 129, 131, 193, 220, 326, 366, 433, 439, 456, 505	2, 4, 6, 8, 10, 17, 18, 19, 22, 25, 46, 49, 50, 54, 55, 57, 66, 74, 82, 122, 145, 146, 147, 152, 196, 210, 217, 239, 257, 258, 260, 261, 266, 273, 274, 275, 277, 279, 281, 289, 290, 293, 296, 298, 317, 321, 337, 345, 361, 385, 386, 388, 391, 449, 458, 467, 512

Доказательство проведено с помощью следующей программы:

Таблица № 3

Программа вычисления номеров 2-нильпотентных и идемпотентных матриц		
№	**Текст программы**	**Коментарии**
1	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)

2	`El:=Elements(A);`	Создание массива элементов алгебры A
3	`NI:=[];`	Создание массива для номеров 2-нильпотентных матриц
4	`ID:=[];`	Создание массива для номеров идемпотентных матриц
5	`for i in [2..512] do`	Начало цикла для проверки ненулевых матриц на 2-нильпотентность или идемпотентность
6	`if El[i]^2=El[1] then`	Проверка на 2-нильпотентность
7	`Add(NI,i);`	Добавление номера 2-нильпотентной матрицы
8	`elif El[i]^2=El[i] then`	Проверка на идемпотентность
9	`Add(ID,i);`	Добавление номера идемпотентной матрицы
10	`fi;`	Конец проверки
11	`od;`	Конец цикла
12	`Print(NI,"\n");`	Печать массива номеров 2-нильпотентных матриц
13	`Print("|NI|=",Size(NI),"\n");`	Печать количества 2-нильпотентных матриц
14	`Print(ID,"\n");`	Печать массива номеров идемпотентных матриц
15	`Print("|ID|=",Size(ID),"\n");`	Печать количества идемпотентных матриц

16	`PrintTo("2gen.dan",`
	`"NI:=",NI,";","\n",`
	`"ID:=",ID,";","\n");`
		Сохранение массива номеров 2-нильпотентных и идемпотентных матриц в файле “2gen.dan”

Результат работы программы выглядит следующим образом:

NI:=[3, 5, 7, 9, 28, 33, 37, 41, 64, 65, 73, 129, 131, 193, 220, 326, 366, 433, 439, 456, 505];

|NI|=21

ID:= [2, 4, 6, 8, 10, 17, 18, 19, 22, 25, 46, 49, 50, 54, 55, 57, 66, 74, 82, 122, 145, 146, 147, 152, 196, 210, 217, 239, 257, 258, 260, 261, 266, 273, 274, 275, 277, 279, 281, 289, 290, 293, 296, 298, 317, 321, 337, 345, 361, 385, 386, 388, 391, 449, 458, 467, 512];

|ID|=57.

Это доказывает теорему.

3.2. Моногенные подалгебры порядка 4

Теорема 3. *В алгебре $M_3(GF(2))$ содержится всего 238 различных моногенных подалгебр порядка 4, из которых*

21 подалгебра порождается нильпотентным элементом индекса 3;

84 подалгебры изоморфны алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=0$;

105 подалгебр изоморфны алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=r$;

28 подалгебр изоморфны полю $GF(2^2)$.

Истинность этой теоремы вытекает из теорем 3.1 – 3.4, доказанных ниже.

3.2.1 Моногенные нильпотентные подалгебры порядка 4

Теорема 3.1. *В алгебре $M_3(GF(2))$ содержится в точности 21 моногенная нильпотентная подалгебра порядка 4. Номера матриц, порождающих такие подалгебры, приведены в таблице № 4.*

Таблица № 4

Номера матриц, порождающих различные четырехэлементные нильпотентные моногенные подалгебры
13, 32, 35, 67, 79, 92, 97, 120, 133, 137, 171, 190, 222, 229, 244, 328, 334, 375, 435, 441, 477
Номера всех матриц, порождающих четырехэлементные нильпотентные моногенные подалгебры
13, 32, 35, 39, 45, 60, 67, 79, 92, 97, 105, 120, 133, 135, 137, 156, 171, 190, 195, 201, 222, 229, 244, 256, 328, 334, 358, 364, 375, 379, 400, 415, 430, 435, 437, 441, 454, 477, 484, 496, 497, 511

Доказательство проведено с помощью следующей программы:

Таблица № 5

Программа вычисления номеров 3-нильпотентных матриц		
№	**Текст программы**	**Коментарии**
1	`Nil:=[];`	Создание массива для номеров всех 3-нильпотентных матриц
2	`Nil3:=[];`	Создание массива для номеров 3-нильпотентных матриц, порождающих различные моногенные подалгебры
3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов алгебры A
5	`for i in [2..512] do`	Начало цикла
6	`if (El[i])^3=Zero(A) and (El[i])^2<>Zero(A) then`	Проверка матриц на 3-нильпотентность
7	`Add(Nil,i);`	Сохранение номера матрицы в массиве Nil
8	`Add(Nil3,i);`	Сохранение номера матрицы в массиве Nil3
9	`fi;`	Конец проверки
10	`od;`	Конец цикла
11	`for i in Nil3 do`	Начало цикла
12	`for j in Nil3 do`	Начало цикла в цикле
13	`if i<j and Subalgebra(A,[El[i]])= Subalgebra(A,[El[j]]) then`	Определение матриц, порождающих одну и ту же моногенную подалгебру
14	`SubtractSet(Nil3,[j]);`	Удаление повторяющих образующих
15	`fi;`	Конец проверки
16	`od;`	Конец цикла в цикле
17	`od;`	Конец цикла
18	`PrintTo("nil_3.dan","Nil3:=",Nil3, ";","\n","|Nil3|=",Size(Nil3),"\n", "Nil:=",Nil,";");`	Сохранение номеров всех 3-нильпотентных матриц и образующих

Результат работы программы выглядит следующим образом:

Nil3:=[13, 32, 35, 67, 79, 92, 97, 120, 133, 137, 171, 190, 222, 229, 244, 328, 334, 375, 435, 441, 477];

|Nil3|=21;

Nil:=[13, 32, 35, 39, 45, 60, 67, 79, 92, 97, 105, 120, 133, 135, 137, 156, 171, 190, 195, 201, 222, 229, 244, 256, 328, 334, 358, 364, 375, 379, 400, 415, 430, 435, 437, 441, 454, 477, 484, 496, 497, 511];

Это доказывает теорему.

3.2.2 Моногенные подалгебры порядка 4, разложимые в прямые суммы

Теорема 3.2. *В алгебре $M_3(GF(2))$ содержится в точности 84 моногенные подалгебры порядка 4, изоморфные алгебре*

$$\langle e\rangle \oplus \langle r\rangle, \text{ где } e^2=e,\ r^2=0,\ er=re=0.$$

Номера матриц, порождающих такие подалгебры, приведены в таблице № 6.

Таблица № 6

Номера матриц, порождающих подалгебры, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e,\ r^2=0,\ er=re=0$
14, 21, 23, 29, 34, 36, 38, 40, 42, 51, 53, 61, 68, 80, 81, 83, 89, 98, 106, 113, 121, 127, 130, 132, 134, 136, 138, 149, 151, 153, 176, 187, 194, 202, 209, 211, 223, 232, 236, 238, 247, 253, 259, 263, 265, 269, 284, 288, 291, 295, 297, 301, 316, 320, 323, 329, 342, 344, 348, 350, 353, 367, 374, 382, 387, 389, 393, 412, 431, 434, 436, 438, 440, 442, 451, 457, 463, 470, 472, 476, 485, 491, 498, 506

Доказательство проведено с помощью следующей программы:

Таблица № 7

Программа, вычисляющая номера матриц, порождающих подалгебры, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e,\ r^2=0,\ er=re=0$		
№	**Текст программы**	**Коментарии**
1	`ER0:=[];`	Создание массива для номеров матриц
2	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
3	`El:=Elements(A);`	Создание массива элементов алгебры A
4	`Read("2gen.dan");`	Чтение файла с данными
5	`for i in ID do`	Начало цикла
6	`for k in NI do`	Начало цикла в цикле
7	`if El[i]*El[k]=El[1] and El[k]*El[i]=El[1] then`	Проверка главного условия: er=re=0
8	`Add(ER0,Position(El,El[i]+El[k]));`	Запись в массив ER0 номера порождающего элемента e+r
9	`fi;`	Конец проверки условия
10	`od;`	Конец цикла в цикле
11	`od;`	Конец цикла
12	`Sort(ER0);`	Упорядочение по возрастанию номеров матриц в массиве ER0
13	`PrintTo("erre0.dan","ER0:=",ER0, ";","\n","|ER0|=",Size(ER0),"\n");`	Сохранение номеров матриц в файле "erre0.dan"

Результат работы программы выглядит следующим образом:

ER0:=[14, 21, 23, 29, 34, 36, 38, 40, 42, 51, 53, 61, 68, 80, 81, 83, 89, 98, 106, 113, 121, 127, 130, 132, 134, 136, 138, 149, 151, 153, 176, 187, 194, 202, 209, 211, 223, 232, 236, 238, 247, 253, 259, 263, 265, 269, 284, 288, 291, 295, 297, 301, 316, 320, 323, 329, 342, 344, 348, 350, 353, 367, 374, 382, 387, 389, 393, 412, 431, 434, 436, 438, 440, 442, 451, 457, 463, 470, 472, 476, 485, 491, 498, 506];

|ER0|=84.

3.2.3 Моногенные подалгебры порядка 4 с единицей, разложимые в прямые суммы

Теорема 3.3. *В алгебре $M_3(GF(2))$ содержится в точности 105 моногенных подалгебр порядка 4, изоморфных алгебре*

$$\langle e\rangle \oplus \langle r\rangle, \text{ где } e^2=e,\ r^2=0,\ er=re=r.$$

Номера матриц, порождающих такие подалгебры, приведены в таблице № 8.

Таблица № 8

Номера матриц, порождающих подалгебры, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e,\ r^2=0,\ er=re=r$
11, 15, 20, 24, 26, 30, 43, 47, 52, 56, 58, 62, 69, 71, 75, 77, 84, 85, 90, 101, 109, 114, 125, 128, 139, 148, 150, 154, 161, 162, 163, 165, 167, 168, 169, 192, 197, 199, 203, 212, 215, 218, 224, 225, 233, 234, 246, 252, 262, 264, 267, 270, 276, 278, 280, 282, 285, 292, 294, 302, 303, 305, 306, 307, 309, 310, 311, 313, 314, 322, 324, 330, 338, 339, 346, 351, 354, 362, 368, 369, 377, 390, 392, 394, 401, 402, 403, 404, 405, 407, 409, 428, 447, 450, 452, 459, 464, 465, 466, 473, 488, 494, 503, 507, 509

Доказательство проведено с помощью следующей программы:

Таблица № 9

Программа, вычисляющая номера матриц, порождающих подалгебры, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e,\ r^2=0,\ er=re=r$		
№	**Текст программы**	**Коментарии**
1	`ERR:=[];`	Создание массива для номеров матриц
2	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)

3	`El:=Elements(A);`	Создание массива элементов алгебры A
4	`Read("2gen.dan");`	Чтение файла с данными
5	`for i in ID do`	Начало цикла
6	`for k in NI do`	Начало цикла в цикле
7	`if El[i]*El[k]=El[k] and El[k]*El[i]=El[k] then`	Проверка главного условия: er=re=r
8	`Add(ERR,Position(El,El[i]+El[k]));`	Запись в массив ERR номера порождающего элемента e + r
9	`fi;`	Конец проверки условия
10	`od;`	Конец цикла в цикле
11	`od;`	Конец цикла
12	`Sort(ERR);`	Упорядочение по возрастанию номеров матриц в массиве ERR
13	`PrintTo("errer.dan","ERR:=",ERR, ";","\n","|ERR|=",Size(ERR),"\n");`	Сохранение номеров матриц в файле "errer.dan"

Результат работы программы выглядит следующим образом:

ERR:= [11, 15, 20, 24, 26, 30, 43, 47, 52, 56, 58, 62, 69, 71, 75, 77, 84, 85, 90, 101, 109, 114, 125, 128, 139, 148, 150, 154, 161, 162, 163, 165, 167, 168, 169, 192, 197, 199, 203, 212, 215, 218, 224, 225, 233, 234, 246, 252, 262, 264, 267, 270, 276, 278, 280, 282, 285, 292, 294, 302, 303, 305, 306, 307, 309, 310, 311, 313, 314, 322, 324, 330, 338, 339, 346, 351, 354, 362, 368, 369, 377, 390, 392, 394, 401, 402, 403, 404, 405, 407, 409, 428, 447, 450, 452, 459, 464, 465, 466, 473, 488, 494, 503, 507, 509];

|ERR|=105

3.2.4 Моногенные подалгебры, изоморфные полю $GF(2^2)$

Теорема 3.4. *В алгебре $M_3(GF(2))$ содержится в точности 28 моногенных подалгебр, изоморфных полю $GF(2^2)$. Номера матриц, порождающих такие подалгебры, приведены в таблице № 10.*

Таблица № 10

Номера матриц, порождающих различные четырехэлементные поля
12, 16, 31, 48, 70, 72, 76, 78, 91, 102, 110, 119, 155, 174, 177, 179, 181, 183, 185, 198, 200, 221, 228, 241, 249, 363, 372, 399
Номера всех матриц, порождающих четырехэлементные поля
12, 16, 27, 31, 44, 48, 59, 63, 70, 72, 76, 78, 91, 102, 110, 119, 140, 155, 174, 177, 179, 181, 183, 185, 198, 200, 204, 219, 221, 228, 241, 249, 325, 327, 333, 357, 363, 365, 372, 384, 399, 414, 417, 419, 421, 423, 425, 448, 453, 455, 480, 481, 489, 504, 508, 510

Доказательство проведено с помощью следующей программы:

Таблица № 11

Программа, вычисляющая номера матриц, порождающих поля порядка 4		
№	**Текст программы**	**Коментарии**
1	`F4:=[];`	Создание массива для номеров матриц, порожда-ющих различные поля порядка 4
2	`F44:=[];`	Создание массива для номеров всех матриц, порождающих поля порядка 4
3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов ал-гебры A
5	`for i in [2..512] do`	Начало цикла
6	`if Size(Subalgebra(A,[El[i]]))=4 and El[i]^4=El[i] and El[i]^2<>El[i] then`	Проверка главного условия: $\|F\|=4$, $a^4=a$, $a^2\neq a$
7	`Add(F4,i);`	Добавление в массив F4 номеров матриц, порожда-ющих поле
8	`Add(F44,i);`	Добавление в массив F44 номеров матриц, порожда-ющих поле
9	`fi;`	Конец проверки главного условия
10	`od;`	Конец цикла
11	`for i in F44 do`	Начало цикла
12	`for j in F44 do`	Начало цикла в цикле
13	`if i<j and Subalgebra(A,[El[i]])= Subalgebra(A,[El[j]]) then`	Определение матриц, порожда-ющих одну и ту же моногенную подалгебру
14	`SubtractSet(F4,[j]);`	Удаление повторяющих образую-щих
15	`fi;`	Конец проверки условия совпадения подалгебр
16	`od;`	Конец цикла в цикле
17	`od;`	Конец цикла
18	`PrintTo("F4.dan","F4:=",F4,";", "\n","|F4|=",Size(F4),"\n", "F44:=",F44,";");`	Сохранение номеров всех матриц, порождающих поле, и номеров образующих

Результат работы программы выглядит следующим образом:

F4:=[12, 16, 31, 48, 70, 72, 76, 78, 91, 102, 110, 119, 155, 174, 177, 179, 181, 183, 185, 198, 200, 221, 228, 241, 249, 363, 372, 399];

|F4|=28

F44:=[12, 16, 27, 31, 44, 48, 59, 63, 70, 72, 76, 78, 91, 102, 110, 119,

140, 155, 174, 177, 179, 181, 183, 185, 198, 200, 204, 219, 221, 228, 241, 249, 325, 327, 333, 357, 363, 365, 372, 384, 399, 414, 417, 419, 421, 423, 425, 448, 453, 455, 480, 481, 489, 504, 508, 510];

3.3. Моногенные подалгебры порядка 8

Теорема 4. *В алгебре $M_3(GF(2))$ содержится всего 57 различных моногенных подалгебр порядка 8, из которых*

8 подалгебр изоморфны полю $GF(2^3)$;

28 подалгебр изоморфны алгебре $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A;

21 подалгебра изоморфна алгебре $\langle r\rangle \oplus\langle e\rangle$, где $r^3=0$, $r^2\neq 0$, e – единица алгебры A.

Истинность этой теоремы вытекает из теорем 4.1 – 4.3, доказанных ниже.

3.3.1 Моногенные подалгебры, изоморфные полю $GF(2^3)$

Теорема 4.1. *В алгебре $M_3(GF(2))$ содержится в точности 8 моногенных подалгебр, изоморфных полю $GF(2^3)$. Номера матриц, порождающих такие подалгебры, приведены в таблице № 12.*

Таблица № 12

Номера матриц, порождающих различные восьмиэлементные поля
95, 96, 100, 103, 107, 112, 115, 116
Номера всех матриц, порождающих восьмиэлементные поля
95, 96, 100, 103, 107, 112, 115, 116, 142, 143, 157, 158, 172, 173, 188, 191, 205, 208, 227, 230, 245, 248, 251, 254, 335, 336, 355, 356, 371, 376, 380, 383, 397, 398, 413, 416, 427, 432, 443, 446, 478, 479, 486, 487, 492, 493, 500, 501

Доказательство проведено с помощью следующей программы:

Таблица № 13

Программа, вычисляющая номера матриц, порождающих поля порядка 8		
№	**Текст программы**	**Коментарии**
1	`F8:=[];`	Создание массива для номеров матриц, порождающих различные поля порядка 8
2	`F88:=[];`	Создание массива для номеров всех матриц, порождающих поля порядка 8

3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов алгебры A
5	`for i in [2..512] do`	Начало цикла
6	`if Size(Subalgebra(A,[El[i]]))=8 and Order(El[i])=7 then`	Проверка главного условия: $\|F\|=8$, $ord(a)=7$
7	`Add(F8,i);`	Добавление в массив F8 номеров матриц, порождающих поле
8	`Add(F88,i);`	Добавление в массив F88 номеров матриц, порождающих поле
9	`fi;`	Конец проверки главного условия
10	`od;`	Конец цикла
11	`for i in F88 do`	Начало цикла
12	`for j in F88 do`	Начало цикла в цикле
13	`if i<j and Subalgebra(A,[El[i]])= Subalgebra(A,[El[j]]) then`	Определение матриц, порождающих одну и ту же моногенную подалгебру
14	`SubtractSet(F8,[j]);`	Удаление повторяющих образующих
15	`fi;`	Конец проверки условия совпадения подалгебр
16	`od;`	Конец цикла в цикле
17	`od;`	Конец цикла
18	`PrintTo("F8.dan","F8=",F8,";", "\n","|F8|=",Size(F8),"\n", "F88:=",F88,";");`	Сохранение номеров всех матриц, порождающих поле порядка 8, и номеров образующих элементов

Результат работы программы выглядит следующим образом:

F8=[95, 96, 100, 103, 107, 112, 115, 116];

|F8|=8

F88:=[95, 96, 100, 103, 107, 112, 115, 116, 142, 143, 157, 158, 172, 173, 188, 191, 205, 208, 227, 230, 245, 248, 251, 254, 335, 336, 355, 356, 371, 376, 380, 383, 397, 398, 413, 416, 427, 432, 443, 446, 478, 479, 486, 487, 492, 493, 500, 501];

3.3.2 Моногенные подалгебры, изоморфные прямой сумме $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A

Теорема 4.2. *В алгебре $M_3(GF(2))$ содержится в точности 28 моногенных подалгебр, изоморфных прямой сумме $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A. Номера матриц, порождающих такие подалгебры, приведены в таблице № 14.*

Таблица № 14

Номера матриц, порождающих моногенные подалгебры, изоморфные прямой сумме $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A
283, 287, 272, 319, 341, 343, 347, 349, 332, 373, 381, 360, 396, 445, 418, 420, 422, 424, 426, 469, 471, 462, 499, 482, 490, 124, 99, 160

Доказательство проведено с помощью следующей программы:

Таблица № 15

Программа, вычисляющая номера матриц, порождающих моногенные подалгебры, изоморфные прямой сумме $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A		
№	**Текст программы**	**Коментарии**
1	`F4E:=[];`	Создание массива для номеров искомых матриц
2	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
3	`El:=Elements(A);`	Создание массива элементов алгебры A
4	`Read("F4.dan");`	Чтение файла с номерами матриц, порождающих четырехэлементные поля
5	`for i in F4 do`	Начало цикла
6	`if (Identity(A) in Subalgebra(A, [El[i]])) = false then`	Проверка отсутствия единичной матрицы в поле
7	`Add(F4E, Position(El,El[i] +Identity(A)));`	Накопление в массиве F4E номеров матриц, порождающих искомые подалгебры
8	`fi;`	Конец проверки условия
9	`od;`	Конец цикла
10	`PrintTo("f4E.dan","F4E:=",F4E,";","\n", "|F4E|=",Size(F4E),"\n");`	Сохранения полученных номеров

Результат работы программы выглядит следующим образом:

F4E:=[283, 287, 272, 319, 341, 343, 347, 349, 332, 373, 381, 360, 396, 445, 418, 420, 422, 424, 426, 469, 471, 462, 499, 482, 490, 124, 99, 160];

|F4E|=28

3.3.3 Моногенные подалгебры, изоморфные прямой сумме $\langle r\rangle\oplus\langle e\rangle$, где $r^3=0$, $r^2\neq 0$, e – единица алгебры A

Теорема 4.3. *В алгебре $M_3(GF(2))$ содержится в точности 21 моногенная подалгебра, изоморфная прямой сумме $\langle r\rangle\oplus\langle e\rangle$, где $r^3=0$, $r^2\neq 0$, e –*

единица алгебры A. Номера матриц, порождающих такие подалгебры, приведены в таблице № 16.

Таблица № 16

Номера матриц, порождающих моногенные подалгебры, изоморфные прямой сумме $\langle r\rangle \oplus \langle e\rangle$, *где* $r^3=0$, $r^2\neq 0$, *e – единица алгебры A*
87, 93, 104, 164, 170, 206, 271, 286, 308, 331, 340, 352, 359, 370, 406, 410, 429, 444, 461, 483, 502

Доказательство проведено с помощью следующей программы:

Таблица № 17

Программа, вычисляющая номера матриц, порождающих моногенные подалгебры, изоморфные прямой сумме $\langle r\rangle \oplus \langle e\rangle$, где $r^3=0$, $r^2\neq 0$, e – единица алгебры A		
№	**Текст программы**	**Коментарии**
1	`R3E:=[];`	Создание массива для номеров искомых матриц
2	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
3	`El:=Elements(A);`	Создание массива элементов алгебры A
4	`Read("nil_3.dan");`	Чтение файла с номерами матриц, порождающих нильпотентные подалгебры порядка 4
5	`for i in Nil3 do`	Начало цикла
6	`Add(R3E, Position(El,El[i]+Identity(A)));`	Накопление в массиве R3E номеров матриц, порождающих искомые подалгебры
7	`od;`	Конец цикла
8	`Sort(R3E);`	Упорядочение номеров
9	`PrintTo("R3E.dan","R3E:=",R3E,";","\n" "|R3E|=",Size(R3E),"\n");`	Сохранение полученных номеров

Результат работы программы выглядит следующим образом:

R3E:=[87, 93, 104, 164, 170, 206, 271, 286, 308, 331, 340, 352, 359, 370,

406, 410, 429, 444, 461, 483, 502];

|R3E|=21

3.3.4 Порождающие элементы алгебры матриц $M_3(GF(2))$

Выясним, какими элементами порождается вся алгебра матриц $M_3(GF(2))$. Прежде всего, представим вместе элементы, порождающие моногенные подалгебры разного типа. Будем, как и прежде, указывать

номера матриц в массиве El, содержащем все 512 элементов алгебры $M_3(GF(2))$. Следующая теорема вытекает из теорем 2, 3.1 – 3.4, 4.1 – 4.3.

Теорема 5. *В таблице № 18 содержатся номера матриц, порождающие все 374 моногенные подалгебры алгебры* $M_3(GF(2))$.

Таблица № 18

№	Название массива	Номера матриц в массиве El	Кол-во матриц
Нулевая матрица			
1		1	1
Нильпотентные подалгебры порядка 2			
2	NI	3, 5, 7, 9, 28, 33, 37, 41, 64, 65, 73, 129, 131, 193, 220, 326, 366, 433, 439, 456, 505	21
Ненильпотентные подалгебры порядка 2			
3	ID	2, 4, 6, 8, 10, 17, 18, 19, 22, 25, 46, 49, 50, 54, 55, 57, 66, 74, 82, 122, 145, 146, 147, 152, 196, 210, 217, 239, 257, 258, 260, 261, 266, 273, 274, 275, 277, 279, 281, 289, 290, 293, 296, 298, 317, 321, 337, 345, 361, 385, 386, 388, 391, 449, 458, 467, 512	57
Нильпотентные моногенные подалгебры четвертого порядка			
4	Nil3	13, 32, 35, 67, 79, 92, 97, 120, 133, 137, 171, 190, 222, 229, 244, 328, 334, 375, 435, 441, 477	21
Моногенные подалгебры четвертого порядка, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=0$			
5	ER0	14, 21, 23, 29, 34, 36, 38, 40, 42, 51, 53, 61, 68, 80, 81, 83, 89, 98, 106, 113, 121, 127, 130, 132, 134, 136, 138, 149, 151, 153, 176, 187, 194, 202, 209, 211, 223, 232, 236, 238, 247, 253, 259, 263, 265, 269, 284, 288, 291, 295, 297, 301, 316, 320, 323, 329, 342, 344, 348, 350, 353, 367, 374, 382, 387, 389, 393, 412, 431, 434, 436, 438, 440, 442, 451, 457, 463, 470, 472, 476, 485, 491, 498, 506	84
Моногенные подалгебры четвертого порядка, изоморфные алгебре $\langle e\rangle \oplus \langle r\rangle$, где $e^2=e$, $r^2=0$, $er=re=r$			
6	ERR	11, 15, 20, 24, 26, 30, 43, 47, 52, 56, 58, 62, 69, 71, 75, 77, 84, 85, 90, 101, 109, 114, 125, 128, 139, 148, 150, 154, 161, 162, 163, 165, 167, 168, 169, 192, 197, 199, 203, 212, 215, 218, 224, 225, 233, 234, 246, 252, 262, 264, 267, 270, 276, 278, 280, 282, 285, 292, 294, 302, 303, 305, 306, 307, 309, 310, 311, 313, 314, 322, 324, 330, 338, 339, 346, 351, 354, 362, 368, 369, 377, 390, 392, 394, 401, 402, 403, 404, 405, 407, 409, 428, 447, 450, 452, 459, 464, 465, 466, 473, 488, 494, 503, 507,	105

		509	
Четырехэлементные поля			
7	F4	12, 16, 31, 48, 70, 72, 76, 78, 91, 102, 110, 119, 155, 174, 177, 179, 181, 183, 185, 198, 200, 221, 228, 241, 249, 363, 372, 399	28
Восьмиэлементные поля			
8	F8	95, 96, 100, 103, 107, 112, 115, 116	8
Моногенные подалгебры, изоморфные прямой сумме $\langle r\rangle \oplus\langle e\rangle$, *где* $r^3=0$, $r^2\neq 0$, e – единица алгебры A			
9	R3E	87, 93, 104, 164, 170, 206, 271, 286, 308, 331, 340, 352, 359, 370, 406, 410, 429, 444, 461, 483, 502	21
Моногенные подалгебры, изоморфные прямой сумме $GF(2^2)\oplus\langle e\rangle$, где e – единица алгебры A			
10	F4E	283, 287, 272, 319, 341, 343, 347, 349, 332, 373, 381, 360, 396, 445, 418, 420, 422, 424, 426, 469, 471, 462, 499, 482, 490, 124, 99, 160	28
		Итого:	**374**

Теорема 6. *Порядок подалгебры, порожденной двумя различными нильпотентными матрицами индекса нильпотентности 2, может быть равен одному из чисел: 4, 8, 16.*

Доказательство проведено с помощью следующей программы:

Таблица № 19

Программа вычисляющая порядки подалгебр, порожденных двумя нильпотентными матрицами индекса 2		
№	**Текст программы**	**Коментарии**
1	`ORD:=[];`	Создание массива для записи порядков
2	`GEN:=[];`	Создание массива для записи номеров
3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов алгебры A
5	`Read("2gen.dan");`	Чтение файла с номерами 2-нильпотентных матриц
6	`for i in NI do`	Начало цикла
7	`for j in NI do`	Начало цикла в цикле
8	`B:=Subalgebra(A,[El[i],El[j]]);`	Построение подалгебры, порожденной двумя элементами
9	`if i<j then`	Исключение

		повторяющихся пар
10	`AddSet(ORD,Size(B));`	Добавление новых порядков подалгебр
11	`if Size(B)=512 then`	Проверка условия порождения всей алгебры A
12	`Add(GEN,[i,j]);`	Добавление пары номеров матриц, порождающих всю алгебру
13	`fi;`	Конец исключения повторяющихся пар
14	`fi;`	Конец проверки порождения всей алгебры
15	`od;`	Конец цикла в цикле
16	`od;`	Конец цикла
17	`PrintTo("gen_2nil.dan","ORD:=",ORD, ";","\n","GEN:=",GEN,";","\n");`	Сохранение результатов вычислений

Результат работы программы: ORD:=[4, 8, 16]; GEN:=[ ];

Теорема 7. *Порядок подалгебры, порожденной двумя различными идемпотентными матрицами, может быть равен одному из чисел: 4, 8, 16, 32.*

Доказательство проведено с помощью следующей программы:

Таблица № 20

Программа вычисляющая порядки подалгебр, порожденных двумя идемпотентными матрицами		
№	**Текст программы**	**Коментарии**
1	`ORD:=[];`	Создание массива для записи порядков
2	`GEN:=[];`	Создание массива для записи номеров
3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов алгебры A
5	`Read("2gen.dan");`	Чтение файла с номерами идемпотентных матриц
6	`for i in NI do`	Начало цикла
7	`for j in NI do`	Начало цикла в цикле
8	`B:=Subalgebra(A,[El[i],El[j]]);`	Построение подалгебры, порожденной двумя элементами
9	`if i<j then`	Исключение повторяющихся пар

10	`AddSet(ORD,Size(B));`	Добавление новых порядков подалгебр
11	`if Size(B)=512 then`	Проверка условия порождения всей алгебры A
12	`Add(GEN,[i,j]);`	Добавление пары номеров матриц, порождающих всю алгебру
13	`fi;`	Конец исключения повторяющихся пар
14	`fi;`	Конец проверки порождения всей алгебры
15	`od;`	Конец цикла в цикле
16	`od;`	Конец цикла
17	`PrintTo("gen_2id.dan","ORD:=",ORD, ";","\n","GEN:=",GEN,";","\n");`	Сохранение результатов вычислений

Результат работы программы: ORD:=[4, 8, 16, 32]; GEN:=[ ];

Теорема 8. *Порядок подалгебры, порожденной двумя различными нильпотентными матрицами индекса нильпотентности 3, может быть равен одному из чисел: 64, 512.*

Доказательство проведено с помощью следующей программы:

Таблица № 21

Программа вычисляющая порядки подалгебр, порожденных двумя нильпотентными матрицами индекса 3		
№	**Текст программы**	**Коментарии**
1	`ORD:=[];`	Создание массива для записи порядков
2	`GEN:=[];`	Создание массива для записи номеров
3	`A:=MatAlgebra(GF(2),3);`	Построение алгебры матриц третьего порядка над полем GF(2)
4	`El:=Elements(A);`	Создание массива элементов алгебры A
5	`Read("nil_3.dan");`	Чтение файла с номерами 3-нильпотентных матриц
6	`for i in Nil3 do`	Начало цикла
7	`for j in Nil3 do`	Начало цикла в цикле
8	`B:=Subalgebra(A,[El[i],El[j]]);`	Построение подалгебры, порожденной двумя элементами
9	`if i<j then`	Исключение повторяющихся пар
10	`AddSet(ORD,Size(B));`	Добавление новых порядков подалгебр

11	`if Size(B)=512 then`	Проверка условия порождения всей алгебры A
12	`Add(GEN,[i,j]);`	Добавление пары номеров матриц, порождающих всю алгебру
13	`fi;`	Конец исключения повторяющихся пар
14	`fi;`	Конец проверки порождения всей алгебры
15	`od;`	Конец цикла в цикле
16	`od;`	Конец цикла
17	`PrintTo("gen_3nil.dan","ORD:=",ORD, ";","\n","GEN:=",GEN,";","\n");`	Сохранение результатов вычислений

Результат работы программы:

ORD:=[64, 512];

GEN:=[[13, 67], [13, 79], [13, 92], [13, 120], [13, 133],
[13, 137], [13, 171], [13, 190], [13, 222], [13, 229],
[13, 244], [13, 328], [13, 375], [13, 435], [13, 441],
[13, 477], [32, 67], [32, 79], [32, 92], [32, 97], [32, 120],
[32, 133], [32, 137], [32, 171], [32, 190], [32, 222],
[32, 244], [32, 328], [32, 334], [32, 375], [32, 435],
[32, 441], [35, 67], [35, 79], [35, 92], [35, 97], [35, 120],
[35, 137], [35, 171], [35, 190], [35, 222], [35, 229],
[35, 244], [35, 328], [35, 334], [35, 375], [35, 441],
[35, 477], [67, 79], [67, 97], [67, 120], [67, 171], [67, 190],
[67, 222], [67, 229], [67, 244], [67, 334], [67, 375],
[67, 435], [67, 441], [67, 477], [79, 92], [79, 97], [79, 133],
[79, 137], [79, 171], [79, 190], [79, 229], [79, 244],
[79, 328], [79, 334], [79, 375], [79, 477], [92, 97],
[92, 120], [92, 133], [92, 171], [92, 190], [92, 222],
[92, 244], [92, 328], [92, 334], [92, 435], [92, 441],
[92, 477], [97, 120], [97, 133], [97, 171], [97, 190],
[97, 222], [97, 229], [97, 244], [97, 328], [97, 375],

[97, 435], [97, 477], [120, 133], [120, 137], [120, 171],
[120, 222], [120, 229], [120, 244], [120, 328], [120, 334],
[120, 375], [120, 435], [133, 137], [133, 171], [133, 190],
[133, 222], [133, 229], [133, 244], [133, 334], [133, 375],
[133, 441], [133, 477], [137, 171], [137, 190], [137, 222],
[137, 229], [137, 244], [137, 328], [137, 334], [137, 375],
[137, 435], [137, 477], [171, 222], [171, 229], [171, 375],
[171, 435], [171, 441], [171, 477], [190, 222], [190, 229],
[190, 244], [190, 334], [190, 375], [190, 435], [190, 441],
[222, 229], [222, 328], [222, 334], [222, 441], [222, 477],
[229, 244], [229, 328], [229, 334], [229, 435], [229, 441],
[244, 328], [244, 435], [244, 441], [244, 477], [328, 334],
[328, 375], [328, 435], [328, 441], [328, 477], [334, 375],
[334, 435], [334, 441], [334, 477], [375, 435], [375, 441],
[375, 477], [435, 441], [435, 477], [441, 477]];

3.4. Решетки подалгебр моногенных алгебр матричной алгебры $M_3(GF(2))$

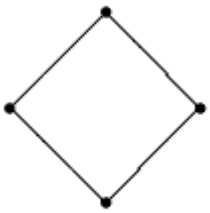
В этой части работы находятся решетки подалгебр всех моногенных алгебр матричной алгебры $M_3(GF(2))$. Ясно, что если решетка подалгебры найдена, то легко определяется и тип решетки. Однако в случае моногенных алгебр по типу решетки нетрудно определить и саму решетку.

Начнем с подалгебр четвертого порядка, так как решетки одно- и двухэлементных подалгебр довольно просты и не требуют специального исследования. Поскольку моногенные подалгебры описаны с точностью до изоморфизма, то достаточно определить решетку подалгебр для одной моногенной подалгебры из каждого класса разбиения.

III

рис. 2

В алгебре $\langle r \rangle$, где $r^3 = 0$, $r^2 \neq 0$, четыре элемента: 0, r, r^2, $r+r^2$, два из которых r и $r+r^2$ порождают саму алгебру $\langle r \rangle$, а элемент r^2 – подалгебру второго порядка. Следовательно, в алгебре $\langle r \rangle$ всего три подалгебры, а значит, ее решетка подалгебр есть решетка III, а ее тип – (1, 1, 1) (см. рис. 2.).

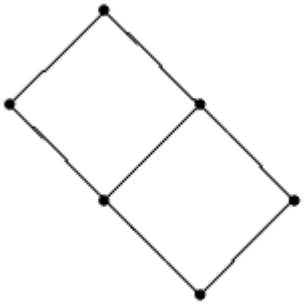


IV

рис. 3

В алгебрах $\langle e \rangle \oplus \langle r \rangle$, где $e^2=e$, $r^2=0$, $er=re=0$, и $\langle e \rangle \oplus \langle r \rangle$, где $e^2=e$, $r^2=0$, $er=re=r$, содержатся по две собственных подалгебры $\langle e \rangle$ и $\langle r \rangle$, а потому решетка подалгебр каждой из них есть решетка IV, а ее тип – (1, 2, 1) (см. рис. 3.).

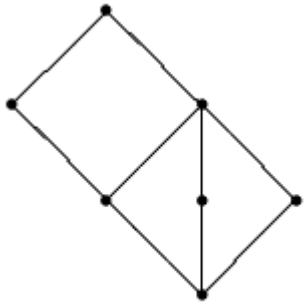
В поле $GF(2^3)$, согласно общей теории, содержится только одно собственное подполе $GF(2)$, а значит, его решетка подалгебр есть решетка III, а ее тип – (1, 1, 0, 1).



V

рис. 4

В алгебре $\langle r\rangle \oplus\langle e\rangle$, *где* $r^3=0$, $r^2\neq 0$, e – единица, всего две подалгебры порядка два: $\langle e\rangle$ и $\langle r^2\rangle$ и две подалгебры порядка четыре: $\langle e, r^2\rangle$ и $\langle r\rangle$. Следовательно, решетка подалгебр этой алгебры есть решетка V, а ее тип – (1, 2, 2, 1) (см. рис. 4.).



VI

рис. 5

В алгебре $GF(2^2)\oplus\langle e\rangle$, *где* e – единица, всего три подалгебры порядка два: $GF(2)$, $\langle e\rangle$ и $\langle e_1 + e\rangle$, где e_1 – единица поля $GF(2^2)$, и две подалгебры по рядка четыре: $GF(2^2)$ и $GF(2)\oplus\langle e\rangle$. Следовательно, решетка подалгебр этой алгебры есть решетка VI, а ее тип – (1, 3, 2, 1) (см. рис. 5.).

Библиографический список

1. Barnes D.W. Lattice isomorphisms of associative algebras //J. Austral. Math. Soc. 1966. V. 6. № 1. P. 106 – 121.
2. Гантмахер Ф.Р. Теория матриц. – 4 изд. – М.: Наука. Гл. ред. физ.-мат. лит. 1988.
3. Биркгоф Г., Барти Т. К. Современная прикладная алгебра; пер. с англ. Ю. И. Манина. – Изд. 2-е, стер. – М.: Лань, 2005. – 400 с.
4. Биркгоф, Г. Теория решеток; пер. с англ. В. Н. Салий под ред. Л. А. Скорнякова. – М.: Наука, 1984. – 568 с.
5. Гретцер, Г. Общая теория решеток; пер. с англ. А. Д. Больбота, В. А. Горбунова, В. И. Туманова под ред. Д. М. Смирнова. – М. : Мир, 1982. – 456 с.
6. Калужнин, Л. А. Введение в общую алгебру. – М.: Наука, 1973. – 448 с.
7. Коробков С. С. Введение в теорию решеток: Учеб.пособие по спец.курсу. Урал.гос.пед.ун-т. — Екатеринбург: Б.и., 1996. – 64с.
8. Курош А. Г. Курс высшей алгебры: Учеб.для студентов вузов по спец." Математика", "Приклад.математика". – 13-е изд., стер. – СПб.: Лань, 2004. – 432с.
9. Курош А. Г. Лекции по общей алгебре: учебник. – СПб.: Лань, 2005. – 560 с.
10. Лидл Р., Пильц Г. Прикладная абстрактная алгебра: Учеб.пособие; Пер.с англ. И.О.Корякова. — Екатеринбург: Изд-во Урал.ун-та, 1996. — 744с.
11. **Система компьютерной алгебры GAP - Exponenta.ru** Режим доступа: www.exponenta.ru/soft/others/gap/1.asp
12. GAP Manual. Режим доступа: http://www.gap-system.org/Doc/manuals.html

Приложение

Массив матриц алгебры A = $M_3(GF(2))$

$m_1=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$, $m_2=\begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}$, $m_3=\begin{pmatrix}0&0&0\\0&0&0\\0&1&0\end{pmatrix}$, $m_4=\begin{pmatrix}0&0&0\\0&0&0\\0&1&1\end{pmatrix}$,

$m_5=\begin{pmatrix}0&0&0\\0&0&0\\1&0&0\end{pmatrix}$, $m_6=\begin{pmatrix}0&0&0\\0&0&0\\1&0&1\end{pmatrix}$, $m_7=\begin{pmatrix}0&0&0\\0&0&0\\1&1&0\end{pmatrix}$, $m_8=\begin{pmatrix}0&0&0\\0&0&0\\1&1&1\end{pmatrix}$,

$m_9=\begin{pmatrix}0&0&0\\0&0&1\\0&0&0\end{pmatrix}$, $m_{10}=\begin{pmatrix}0&0&0\\0&0&1\\0&0&1\end{pmatrix}$, $m_{11}=\begin{pmatrix}0&0&0\\0&0&1\\0&1&0\end{pmatrix}$, $m_{12}=\begin{pmatrix}0&0&0\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{13}=\begin{pmatrix}0&0&0\\0&0&1\\1&0&0\end{pmatrix}$, $m_{14}=\begin{pmatrix}0&0&0\\0&0&1\\1&0&1\end{pmatrix}$, $m_{15}=\begin{pmatrix}0&0&0\\0&0&1\\1&1&0\end{pmatrix}$, $m_{16}=\begin{pmatrix}0&0&0\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{17}=\begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix}$, $m_{18}=\begin{pmatrix}0&0&0\\0&1&0\\0&0&1\end{pmatrix}$, $m_{19}=\begin{pmatrix}0&0&0\\0&1&0\\0&1&0\end{pmatrix}$, $m_{20}=\begin{pmatrix}0&0&0\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{21}=\begin{pmatrix}0&0&0\\0&1&0\\1&0&0\end{pmatrix}$, $m_{22}=\begin{pmatrix}0&0&0\\0&1&0\\1&0&1\end{pmatrix}$, $m_{23}=\begin{pmatrix}0&0&0\\0&1&0\\1&1&0\end{pmatrix}$, $m_{24}=\begin{pmatrix}0&0&0\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{25}=\begin{pmatrix}0&0&0\\0&1&1\\0&0&0\end{pmatrix}$, $m_{26}=\begin{pmatrix}0&0&0\\0&1&1\\0&0&1\end{pmatrix}$, $m_{27}=\begin{pmatrix}0&0&0\\0&1&1\\0&1&0\end{pmatrix}$, $m_{28}=\begin{pmatrix}0&0&0\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{29}=\begin{pmatrix}0&0&0\\0&1&1\\1&0&0\end{pmatrix}$, $m_{30}=\begin{pmatrix}0&0&0\\0&1&1\\1&0&1\end{pmatrix}$, $m_{31}=\begin{pmatrix}0&0&0\\0&1&1\\1&1&0\end{pmatrix}$, $m_{32}=\begin{pmatrix}0&0&0\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{33}=\begin{pmatrix}0&0&0\\1&0&0\\0&0&0\end{pmatrix}$, $m_{34}=\begin{pmatrix}0&0&0\\1&0&0\\0&0&1\end{pmatrix}$, $m_{35}=\begin{pmatrix}0&0&0\\1&0&0\\0&1&0\end{pmatrix}$, $m_{36}=\begin{pmatrix}0&0&0\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{37}=\begin{pmatrix}0&0&0\\1&0&0\\1&0&0\end{pmatrix}$, $m_{38}=\begin{pmatrix}0&0&0\\1&0&0\\1&0&1\end{pmatrix}$, $m_{39}=\begin{pmatrix}0&0&0\\1&0&0\\1&1&0\end{pmatrix}$, $m_{40}=\begin{pmatrix}0&0&0\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{41}=\begin{pmatrix}0&0&0\\1&0&1\\0&0&0\end{pmatrix}$, $m_{42}=\begin{pmatrix}0&0&0\\1&0&1\\0&0&1\end{pmatrix}$, $m_{43}=\begin{pmatrix}0&0&0\\1&0&1\\0&1&0\end{pmatrix}$, $m_{44}=\begin{pmatrix}0&0&0\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{45}=\begin{pmatrix}0&0&0\\1&0&1\\1&0&0\end{pmatrix}$, $m_{46}=\begin{pmatrix}0&0&0\\1&0&1\\1&0&1\end{pmatrix}$, $m_{47}=\begin{pmatrix}0&0&0\\1&0&1\\1&1&0\end{pmatrix}$, $m_{48}=\begin{pmatrix}0&0&0\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{49}=\begin{pmatrix}0&0&0\\1&1&0\\0&0&0\end{pmatrix}$, $m_{50}=\begin{pmatrix}0&0&0\\1&1&0\\0&0&1\end{pmatrix}$, $m_{51}=\begin{pmatrix}0&0&0\\1&1&0\\0&1&0\end{pmatrix}$, $m_{52}=\begin{pmatrix}0&0&0\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{53}=\begin{pmatrix}0&0&0\\1&1&0\\1&0&0\end{pmatrix}$, $m_{54}=\begin{pmatrix}0&0&0\\1&1&0\\1&0&1\end{pmatrix}$, $m_{55}=\begin{pmatrix}0&0&0\\1&1&0\\1&1&0\end{pmatrix}$, $m_{56}=\begin{pmatrix}0&0&0\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{57}=\begin{pmatrix}0&0&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{58}=\begin{pmatrix}0&0&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{59}=\begin{pmatrix}0&0&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{60}=\begin{pmatrix}0&0&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{61}=\begin{pmatrix}0&0&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{62}=\begin{pmatrix}0&0&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{63}=\begin{pmatrix}0&0&0\\1&1&1\\1&1&0\end{pmatrix}$, $m_{64}=\begin{pmatrix}0&0&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{65}=\begin{pmatrix}0&0&1\\0&0&0\\0&0&0\end{pmatrix}$, $m_{66}=\begin{pmatrix}0&0&1\\0&0&0\\0&0&1\end{pmatrix}$, $m_{67}=\begin{pmatrix}0&0&1\\0&0&0\\0&1&0\end{pmatrix}$, $m_{68}=\begin{pmatrix}0&0&1\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{69}=\begin{pmatrix}0&0&1\\0&0&0\\1&0&0\end{pmatrix}$, $m_{70}=\begin{pmatrix}0&0&1\\0&0&0\\1&0&1\end{pmatrix}$, $m_{71}=\begin{pmatrix}0&0&1\\0&0&0\\1&1&0\end{pmatrix}$, $m_{72}=\begin{pmatrix}0&0&1\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{73}=\begin{pmatrix}0&0&1\\0&0&1\\0&0&0\end{pmatrix}$, $m_{74}=\begin{pmatrix}0&0&1\\0&0&1\\0&0&1\end{pmatrix}$, $m_{75}=\begin{pmatrix}0&0&1\\0&0&1\\0&1&0\end{pmatrix}$, $m_{76}=\begin{pmatrix}0&0&1\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{77}=\begin{pmatrix}0&0&1\\0&0&1\\1&0&0\end{pmatrix}$, $m_{78}=\begin{pmatrix}0&0&1\\0&0&1\\1&0&1\end{pmatrix}$, $m_{79}=\begin{pmatrix}0&0&1\\0&0&1\\1&1&0\end{pmatrix}$, $m_{80}=\begin{pmatrix}0&0&1\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{81}=\begin{pmatrix}0&0&1\\0&1&0\\0&0&0\end{pmatrix}$, $m_{82}=\begin{pmatrix}0&0&1\\0&1&0\\0&0&1\end{pmatrix}$, $m_{83}=\begin{pmatrix}0&0&1\\0&1&0\\0&1&0\end{pmatrix}$, $m_{84}=\begin{pmatrix}0&0&1\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{85}=\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}$, $m_{86}=\begin{pmatrix}0&0&1\\0&1&0\\1&0&1\end{pmatrix}$, $m_{87}=\begin{pmatrix}0&0&1\\0&1&0\\1&1&0\end{pmatrix}$, $m_{88}=\begin{pmatrix}0&0&1\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{89}=\begin{pmatrix}0&0&1\\0&1&1\\0&0&0\end{pmatrix}$, $m_{90}=\begin{pmatrix}0&0&1\\0&1&1\\0&0&1\end{pmatrix}$, $m_{91}=\begin{pmatrix}0&0&1\\0&1&1\\0&1&0\end{pmatrix}$, $m_{92}=\begin{pmatrix}0&0&1\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{93}=\begin{pmatrix}0&0&1\\0&1&1\\1&0&0\end{pmatrix}$, $m_{94}=\begin{pmatrix}0&0&1\\0&1&1\\1&0&1\end{pmatrix}$, $m_{95}=\begin{pmatrix}0&0&1\\0&1&1\\1&1&0\end{pmatrix}$, $m_{96}=\begin{pmatrix}0&0&1\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{97}=\begin{pmatrix}0&0&1\\1&0&0\\0&0&0\end{pmatrix}$, $m_{98}=\begin{pmatrix}0&0&1\\1&0&0\\0&0&1\end{pmatrix}$, $m_{99}=\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}$, $m_{100}=\begin{pmatrix}0&0&1\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{101}=\begin{pmatrix}0&0&1\\1&0&0\\1&0&0\end{pmatrix}$, $m_{102}=\begin{pmatrix}0&0&1\\1&0&0\\1&0&1\end{pmatrix}$, $m_{103}=\begin{pmatrix}0&0&1\\1&0&0\\1&1&0\end{pmatrix}$, $m_{104}=\begin{pmatrix}0&0&1\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{105}=\begin{pmatrix}0&0&1\\1&0&1\\0&0&0\end{pmatrix}$, $m_{106}=\begin{pmatrix}0&0&1\\1&0&1\\0&0&1\end{pmatrix}$, $m_{107}=\begin{pmatrix}0&0&1\\1&0&1\\0&1&0\end{pmatrix}$, $m_{108}=\begin{pmatrix}0&0&1\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{109}=\begin{pmatrix}0&0&1\\1&0&1\\1&0&0\end{pmatrix}$, $m_{110}=\begin{pmatrix}0&0&1\\1&0&1\\1&0&1\end{pmatrix}$, $m_{111}=\begin{pmatrix}0&0&1\\1&0&1\\1&1&0\end{pmatrix}$, $m_{112}=\begin{pmatrix}0&0&1\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{113}=\begin{pmatrix}0&0&1\\1&1&0\\0&0&0\end{pmatrix}$, $m_{114}=\begin{pmatrix}0&0&1\\1&1&0\\0&0&1\end{pmatrix}$, $m_{115}=\begin{pmatrix}0&0&1\\1&1&0\\0&1&0\end{pmatrix}$, $m_{116}=\begin{pmatrix}0&0&1\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{117}=\begin{pmatrix}0&0&1\\1&1&0\\1&0&0\end{pmatrix}$, $m_{118}=\begin{pmatrix}0&0&1\\1&1&0\\1&0&1\end{pmatrix}$, $m_{119}=\begin{pmatrix}0&0&1\\1&1&0\\1&1&0\end{pmatrix}$, $m_{120}=\begin{pmatrix}0&0&1\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{121}=\begin{pmatrix}0&0&1\\1&1&1\\0&0&0\end{pmatrix}$, $m_{122}=\begin{pmatrix}0&0&1\\1&1&1\\0&0&1\end{pmatrix}$, $m_{123}=\begin{pmatrix}0&0&1\\1&1&1\\0&1&0\end{pmatrix}$, $m_{124}=\begin{pmatrix}0&0&1\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{125}=\begin{pmatrix}0&0&1\\1&1&1\\1&0&0\end{pmatrix}$, $m_{126}=\begin{pmatrix}0&0&1\\1&1&1\\1&0&1\end{pmatrix}$, $m_{127}=\begin{pmatrix}0&0&1\\1&1&1\\1&1&0\end{pmatrix}$, $m_{128}=\begin{pmatrix}0&0&1\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{129}=\begin{pmatrix}0&1&0\\0&0&0\\0&0&0\end{pmatrix}$, $m_{130}=\begin{pmatrix}0&1&0\\0&0&0\\0&0&1\end{pmatrix}$, $m_{131}=\begin{pmatrix}0&1&0\\0&0&0\\0&1&0\end{pmatrix}$, $m_{132}=\begin{pmatrix}0&1&0\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{133}=\begin{pmatrix}0&1&0\\0&0&0\\1&0&0\end{pmatrix}$, $m_{134}=\begin{pmatrix}0&1&0\\0&0&0\\1&0&1\end{pmatrix}$, $m_{135}=\begin{pmatrix}0&1&0\\0&0&0\\1&1&0\end{pmatrix}$, $m_{136}=\begin{pmatrix}0&1&0\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{137}=\begin{pmatrix}0&1&0\\0&0&1\\0&0&0\end{pmatrix}$, $m_{138}=\begin{pmatrix}0&1&0\\0&0&1\\0&0&1\end{pmatrix}$, $m_{139}=\begin{pmatrix}0&1&0\\0&0&1\\0&1&0\end{pmatrix}$, $m_{140}=\begin{pmatrix}0&1&0\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{141}=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}$, $m_{142}=\begin{pmatrix}0&1&0\\0&0&1\\1&0&1\end{pmatrix}$, $m_{143}=\begin{pmatrix}0&1&0\\0&0&1\\1&1&0\end{pmatrix}$, $m_{144}=\begin{pmatrix}0&1&0\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{145}=\begin{pmatrix}0&1&0\\0&1&0\\0&0&0\end{pmatrix}$, $m_{146}=\begin{pmatrix}0&1&0\\0&1&0\\0&0&1\end{pmatrix}$, $m_{147}=\begin{pmatrix}0&1&0\\0&1&0\\0&1&0\end{pmatrix}$, $m_{148}=\begin{pmatrix}0&1&0\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{149}=\begin{pmatrix}0&1&0\\0&1&0\\1&0&0\end{pmatrix}$, $m_{150}=\begin{pmatrix}0&1&0\\0&1&0\\1&0&1\end{pmatrix}$, $m_{151}=\begin{pmatrix}0&1&0\\0&1&0\\1&1&0\end{pmatrix}$, $m_{152}=\begin{pmatrix}0&1&0\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{153}=\begin{pmatrix}0&1&0\\0&1&1\\0&0&0\end{pmatrix}$, $m_{154}=\begin{pmatrix}0&1&0\\0&1&1\\0&0&1\end{pmatrix}$, $m_{155}=\begin{pmatrix}0&1&0\\0&1&1\\0&1&0\end{pmatrix}$, $m_{156}=\begin{pmatrix}0&1&0\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{157}=\begin{pmatrix}0&1&0\\0&1&1\\1&0&0\end{pmatrix}$, $m_{158}=\begin{pmatrix}0&1&0\\0&1&1\\1&0&1\end{pmatrix}$, $m_{159}=\begin{pmatrix}0&1&0\\0&1&1\\1&1&0\end{pmatrix}$, $m_{160}=\begin{pmatrix}0&1&0\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{161}=\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix}$, $m_{162}=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}$, $m_{163}=\begin{pmatrix}0&1&0\\1&0&0\\0&1&0\end{pmatrix}$, $m_{164}=\begin{pmatrix}0&1&0\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{165}=\begin{pmatrix}0&1&0\\1&0&0\\1&0&0\end{pmatrix}$, $m_{166}=\begin{pmatrix}0&1&0\\1&0&0\\1&0&1\end{pmatrix}$, $m_{167}=\begin{pmatrix}0&1&0\\1&0&0\\1&1&0\end{pmatrix}$, $m_{168}=\begin{pmatrix}0&1&0\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{169}=\begin{pmatrix}0&1&0\\1&0&1\\0&0&0\end{pmatrix}$, $m_{170}=\begin{pmatrix}0&1&0\\1&0&1\\0&0&1\end{pmatrix}$, $m_{171}=\begin{pmatrix}0&1&0\\1&0&1\\0&1&0\end{pmatrix}$, $m_{172}=\begin{pmatrix}0&1&0\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{173}=\begin{pmatrix}0&1&0\\1&0&1\\1&0&0\end{pmatrix}$, $m_{174}=\begin{pmatrix}0&1&0\\1&0&1\\1&0&1\end{pmatrix}$, $m_{175}=\begin{pmatrix}0&1&0\\1&0&1\\1&1&0\end{pmatrix}$, $m_{176}=\begin{pmatrix}0&1&0\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{177}=\begin{pmatrix}0&1&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{178}=\begin{pmatrix}0&1&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{179}=\begin{pmatrix}0&1&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{180}=\begin{pmatrix}0&1&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{181}=\begin{pmatrix}0&1&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{182}=\begin{pmatrix}0&1&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{183}=\begin{pmatrix}0&1&9\\1&1&1\\1&1&0\end{pmatrix}$, $m_{184}=\begin{pmatrix}0&1&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{185}=\begin{pmatrix}0&1&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{186}=\begin{pmatrix}0&1&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{187}=\begin{pmatrix}0&1&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{188}=\begin{pmatrix}0&1&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{189}=\begin{pmatrix}0&1&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{190}=\begin{pmatrix}0&1&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{191}=\begin{pmatrix}0&1&0\\1&1&1\\1&1&0\end{pmatrix}$, $m_{192}=\begin{pmatrix}0&1&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{193}=\begin{pmatrix}0&1&1\\0&0&0\\0&0&0\end{pmatrix}$, $m_{194}=\begin{pmatrix}0&1&1\\0&0&0\\0&0&1\end{pmatrix}$, $m_{195}=\begin{pmatrix}0&1&1\\0&0&0\\0&1&0\end{pmatrix}$, $m_{196}=\begin{pmatrix}0&1&1\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{197}=\begin{pmatrix}0&1&1\\0&0&0\\1&0&0\end{pmatrix}$, $m_{198}=\begin{pmatrix}0&1&1\\0&0&0\\1&0&1\end{pmatrix}$, $m_{199}=\begin{pmatrix}0&1&1\\0&0&0\\1&1&0\end{pmatrix}$, $m_{200}=\begin{pmatrix}0&1&1\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{201}=\begin{pmatrix}0&1&1\\0&0&1\\0&0&0\end{pmatrix}$, $m_{202}=\begin{pmatrix}0&1&1\\0&0&1\\0&0&1\end{pmatrix}$, $m_{203}=\begin{pmatrix}0&1&1\\0&0&1\\0&1&0\end{pmatrix}$, $m_{204}=\begin{pmatrix}0&1&1\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{205}=\begin{pmatrix}0&1&1\\0&0&1\\1&0&0\end{pmatrix}$, $m_{206}=\begin{pmatrix}0&1&1\\0&0&1\\1&0&1\end{pmatrix}$, $m_{207}=\begin{pmatrix}0&1&1\\0&0&1\\1&1&0\end{pmatrix}$, $m_{208}=\begin{pmatrix}0&1&1\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{209}=\begin{pmatrix}0&1&1\\0&1&0\\0&0&0\end{pmatrix}$, $m_{210}=\begin{pmatrix}0&1&1\\0&1&0\\0&0&1\end{pmatrix}$, $m_{211}=\begin{pmatrix}0&1&1\\0&1&0\\0&1&0\end{pmatrix}$, $m_{212}=\begin{pmatrix}0&1&1\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{213}=\begin{pmatrix}0&1&1\\0&1&0\\1&0&0\end{pmatrix}$, $m_{214}=\begin{pmatrix}0&1&1\\0&1&0\\1&0&1\end{pmatrix}$, $m_{215}=\begin{pmatrix}0&1&1\\0&1&0\\1&1&0\end{pmatrix}$, $m_{216}=\begin{pmatrix}0&1&1\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{217}=\begin{pmatrix}0&1&1\\0&1&1\\0&0&0\end{pmatrix}$, $m_{218}=\begin{pmatrix}0&1&1\\0&1&1\\0&0&1\end{pmatrix}$, $m_{219}=\begin{pmatrix}0&1&1\\0&1&1\\0&1&0\end{pmatrix}$, $m_{220}=\begin{pmatrix}0&1&1\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{221}=\begin{pmatrix}0&1&1\\0&1&1\\1&0&0\end{pmatrix}$, $m_{222}=\begin{pmatrix}0&1&1\\0&1&1\\1&0&1\end{pmatrix}$, $m_{223}=\begin{pmatrix}0&1&1\\0&1&1\\1&1&0\end{pmatrix}$, $m_{224}=\begin{pmatrix}0&1&1\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{225}=\begin{pmatrix}0&1&1\\1&0&0\\0&0&0\end{pmatrix}$, $m_{226}=\begin{pmatrix}0&1&1\\1&0&0\\0&0&1\end{pmatrix}$, $m_{227}=\begin{pmatrix}0&1&1\\1&0&0\\0&1&0\end{pmatrix}$, $m_{228}=\begin{pmatrix}0&1&1\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{229}=\begin{pmatrix}0&1&1\\1&0&0\\1&0&0\end{pmatrix}$, $m_{230}=\begin{pmatrix}0&1&1\\1&0&0\\1&0&1\end{pmatrix}$, $m_{231}=\begin{pmatrix}0&1&1\\1&0&0\\1&1&0\end{pmatrix}$, $m_{232}=\begin{pmatrix}0&1&1\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{233}=\begin{pmatrix}0&1&1\\1&0&1\\0&0&0\end{pmatrix}$, $m_{234}=\begin{pmatrix}0&1&1\\1&0&1\\0&0&1\end{pmatrix}$, $m_{235}=\begin{pmatrix}0&1&1\\1&0&1\\0&1&0\end{pmatrix}$, $m_{236}=\begin{pmatrix}0&1&1\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{237}=\begin{pmatrix}0&1&1\\1&0&1\\1&0&0\end{pmatrix}$, $m_{238}=\begin{pmatrix}0&1&1\\1&0&1\\1&0&1\end{pmatrix}$, $m_{239}=\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}$, $m_{240}=\begin{pmatrix}0&1&1\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{241}=\begin{pmatrix}0&1&1\\1&1&0\\0&0&0\end{pmatrix}$, $m_{242}=\begin{pmatrix}0&1&1\\1&1&0\\0&0&1\end{pmatrix}$, $m_{243}=\begin{pmatrix}0&1&1\\1&1&0\\0&1&0\end{pmatrix}$, $m_{244}=\begin{pmatrix}0&1&1\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{245}=\begin{pmatrix}0&1&1\\1&1&0\\1&0&0\end{pmatrix}$, $m_{246}=\begin{pmatrix}0&1&1\\1&1&0\\1&0&1\end{pmatrix}$, $m_{247}=\begin{pmatrix}0&1&1\\1&1&0\\1&1&0\end{pmatrix}$, $m_{248}=\begin{pmatrix}0&1&1\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{249}=\begin{pmatrix}0&1&1\\1&1&1\\0&0&0\end{pmatrix}$, $m_{250}=\begin{pmatrix}0&1&1\\1&1&1\\0&0&1\end{pmatrix}$, $m_{251}=\begin{pmatrix}0&1&1\\1&1&1\\0&1&0\end{pmatrix}$, $m_{252}=\begin{pmatrix}0&1&1\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{253}=\begin{pmatrix}0&1&1\\1&1&1\\1&0&0\end{pmatrix}$, $m_{254}=\begin{pmatrix}0&1&1\\1&1&1\\1&0&1\end{pmatrix}$, $m_{255}=\begin{pmatrix}0&1&1\\1&1&1\\1&1&0\end{pmatrix}$, $m_{256}=\begin{pmatrix}0&1&1\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{257}=\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}$, $m_{258}=\begin{pmatrix}1&0&0\\0&0&0\\0&0&1\end{pmatrix}$, $m_{259}=\begin{pmatrix}1&0&0\\0&0&0\\0&1&0\end{pmatrix}$, $m_{260}=\begin{pmatrix}1&0&0\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{261}=\begin{pmatrix}1&0&0\\0&0&0\\1&0&0\end{pmatrix}$, $m_{262}=\begin{pmatrix}1&0&0\\0&0&0\\1&0&1\end{pmatrix}$, $m_{263}=\begin{pmatrix}1&0&0\\0&0&0\\1&1&0\end{pmatrix}$, $m_{264}=\begin{pmatrix}1&0&0\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{265}=\begin{pmatrix}1&0&0\\0&0&1\\0&0&0\end{pmatrix}$, $m_{266}=\begin{pmatrix}1&0&0\\0&0&1\\0&0&1\end{pmatrix}$, $m_{267}=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}$, $m_{268}=\begin{pmatrix}1&0&0\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{269}=\begin{pmatrix}1&0&0\\0&0&1\\1&0&0\end{pmatrix}$, $m_{270}=\begin{pmatrix}1&0&0\\0&0&1\\1&0&1\end{pmatrix}$, $m_{271}=\begin{pmatrix}1&0&0\\0&0&1\\1&1&0\end{pmatrix}$, $m_{272}=\begin{pmatrix}1&0&0\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{273}=\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}$, $m_{274}=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$, $m_{275}=\begin{pmatrix}1&0&0\\0&1&0\\0&1&0\end{pmatrix}$, $m_{276}=\begin{pmatrix}1&0&0\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{277}=\begin{pmatrix}1&0&0\\0&1&0\\1&0&0\end{pmatrix}$, $m_{278}=\begin{pmatrix}1&0&0\\0&1&0\\1&0&1\end{pmatrix}$, $m_{279}=\begin{pmatrix}1&0&0\\0&1&0\\1&1&0\end{pmatrix}$, $m_{280}=\begin{pmatrix}1&0&0\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{281}=\begin{pmatrix}1&0&0\\0&1&1\\0&0&0\end{pmatrix}$, $m_{282}=\begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix}$, $m_{283}=\begin{pmatrix}1&0&0\\0&1&1\\0&1&0\end{pmatrix}$, $m_{284}=\begin{pmatrix}1&0&0\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{285}=\begin{pmatrix}1&0&0\\0&1&1\\1&0&0\end{pmatrix}$, $m_{286}=\begin{pmatrix}1&0&0\\0&1&1\\1&0&1\end{pmatrix}$, $m_{287}=\begin{pmatrix}1&0&0\\0&1&1\\1&1&0\end{pmatrix}$, $m_{288}=\begin{pmatrix}1&0&0\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{289}=\begin{pmatrix}1&0&0\\1&0&0\\0&0&0\end{pmatrix}$, $m_{290}=\begin{pmatrix}1&0&0\\1&0&0\\0&0&1\end{pmatrix}$, $m_{291}=\begin{pmatrix}1&0&0\\1&0&0\\0&1&0\end{pmatrix}$, $m_{292}=\begin{pmatrix}1&0&0\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{293}=\begin{pmatrix}1&0&0\\1&0&0\\1&0&0\end{pmatrix}$, $m_{294}=\begin{pmatrix}1&0&0\\1&0&0\\1&0&1\end{pmatrix}$, $m_{295}=\begin{pmatrix}1&0&0\\1&0&0\\1&1&0\end{pmatrix}$, $m_{296}=\begin{pmatrix}1&0&0\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{297}=\begin{pmatrix}1&0&0\\1&0&1\\0&0&0\end{pmatrix}$, $m_{298}=\begin{pmatrix}1&0&0\\1&0&1\\0&0&1\end{pmatrix}$, $m_{299}=\begin{pmatrix}1&0&0\\1&0&1\\0&1&0\end{pmatrix}$, $m_{300}=\begin{pmatrix}1&0&0\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{301}=\begin{pmatrix}1&0&0\\1&0&1\\1&0&0\end{pmatrix}$, $m_{302}=\begin{pmatrix}1&0&0\\1&0&1\\1&0&1\end{pmatrix}$, $m_{303}=\begin{pmatrix}1&0&0\\1&0&1\\1&1&0\end{pmatrix}$, $m_{304}=\begin{pmatrix}1&0&0\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{305}=\begin{pmatrix}1&0&0\\1&1&0\\0&0&0\end{pmatrix}$, $m_{306}=\begin{pmatrix}1&0&0\\1&1&0\\0&0&1\end{pmatrix}$, $m_{307}=\begin{pmatrix}1&0&0\\1&1&0\\0&1&0\end{pmatrix}$, $m_{308}=\begin{pmatrix}1&0&0\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{309}=\begin{pmatrix}1&0&0\\1&1&0\\1&0&0\end{pmatrix}$, $m_{310}=\begin{pmatrix}1&0&0\\1&1&0\\1&0&1\end{pmatrix}$, $m_{311}=\begin{pmatrix}1&0&0\\1&1&0\\1&1&0\end{pmatrix}$, $m_{312}=\begin{pmatrix}1&0&0\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{313}=\begin{pmatrix}1&0&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{314}=\begin{pmatrix}1&0&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{315}=\begin{pmatrix}1&0&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{316}=\begin{pmatrix}1&0&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{317}=\begin{pmatrix}1&0&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{318}=\begin{pmatrix}1&0&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{319}=\begin{pmatrix}1&0&0\\1&1&1\\1&1&0\end{pmatrix}$, $m_{320}=\begin{pmatrix}1&0&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{321}=\begin{pmatrix}1&0&1\\0&0&0\\0&0&0\end{pmatrix}$, $m_{322}=\begin{pmatrix}1&0&1\\0&0&0\\0&0&1\end{pmatrix}$, $m_{323}=\begin{pmatrix}1&0&1\\0&0&0\\0&1&0\end{pmatrix}$, $m_{324}=\begin{pmatrix}1&0&1\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{325}=\begin{pmatrix}1&0&1\\0&0&0\\1&0&0\end{pmatrix}$, $m_{326}=\begin{pmatrix}1&0&1\\0&0&0\\1&0&1\end{pmatrix}$, $m_{327}=\begin{pmatrix}1&0&1\\0&0&0\\1&1&0\end{pmatrix}$, $m_{328}=\begin{pmatrix}1&0&1\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{329}=\begin{pmatrix}1&0&1\\0&0&1\\0&0&0\end{pmatrix}$, $m_{330}=\begin{pmatrix}1&0&1\\0&0&1\\0&0&1\end{pmatrix}$, $m_{331}=\begin{pmatrix}1&0&1\\0&0&1\\0&1&0\end{pmatrix}$, $m_{332}=\begin{pmatrix}1&0&1\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{333}=\begin{pmatrix}1&0&1\\0&0&1\\1&0&0\end{pmatrix}$, $m_{334}=\begin{pmatrix}1&0&1\\0&0&1\\1&0&1\end{pmatrix}$, $m_{335}=\begin{pmatrix}1&0&1\\0&0&1\\1&1&0\end{pmatrix}$, $m_{336}=\begin{pmatrix}1&0&1\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{337}=\begin{pmatrix}1&0&1\\0&1&0\\0&0&0\end{pmatrix}$, $m_{338}=\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\end{pmatrix}$, $m_{339}=\begin{pmatrix}1&0&1\\0&1&0\\0&1&0\end{pmatrix}$, $m_{340}=\begin{pmatrix}1&0&1\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{341}=\begin{pmatrix}1&0&1\\0&1&0\\1&0&0\end{pmatrix}$, $m_{342}=\begin{pmatrix}1&0&1\\0&1&0\\1&0&1\end{pmatrix}$, $m_{343}=\begin{pmatrix}1&0&1\\0&1&0\\1&1&0\end{pmatrix}$, $m_{344}=\begin{pmatrix}1&0&1\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{345}=\begin{pmatrix}1&0&1\\0&1&1\\0&0&0\end{pmatrix}$, $m_{346}=\begin{pmatrix}1&0&1\\0&1&1\\0&0&1\end{pmatrix}$, $m_{347}=\begin{pmatrix}1&0&1\\0&1&1\\0&1&0\end{pmatrix}$, $m_{348}=\begin{pmatrix}1&0&1\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{349}=\begin{pmatrix}1&0&1\\0&1&1\\1&0&0\end{pmatrix}$, $m_{350}=\begin{pmatrix}1&0&1\\0&1&1\\1&0&1\end{pmatrix}$, $m_{351}=\begin{pmatrix}1&0&1\\0&1&1\\1&1&0\end{pmatrix}$, $m_{352}=\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{353}=\begin{pmatrix}1&0&1\\1&0&0\\0&0&0\end{pmatrix}$, $m_{354}=\begin{pmatrix}1&0&1\\1&0&0\\0&0&1\end{pmatrix}$, $m_{355}=\begin{pmatrix}1&0&1\\1&0&0\\0&1&0\end{pmatrix}$, $m_{356}=\begin{pmatrix}1&0&1\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{357}=\begin{pmatrix}1&0&1\\1&0&0\\1&0&0\end{pmatrix}$, $m_{358}=\begin{pmatrix}1&0&1\\1&0&0\\1&0&1\end{pmatrix}$, $m_{359}=\begin{pmatrix}1&0&1\\1&0&0\\1&1&0\end{pmatrix}$, $m_{360}=\begin{pmatrix}1&0&1\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{361}=\begin{pmatrix}1&0&1\\1&0&1\\0&0&0\end{pmatrix}$, $m_{362}=\begin{pmatrix}1&0&1\\1&0&1\\0&0&1\end{pmatrix}$, $m_{363}=\begin{pmatrix}1&0&1\\1&0&1\\0&1&0\end{pmatrix}$, $m_{364}=\begin{pmatrix}1&0&1\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{365}=\begin{pmatrix}1&0&1\\1&0&1\\1&0&0\end{pmatrix}$, $m_{366}=\begin{pmatrix}1&0&1\\1&0&1\\1&0&1\end{pmatrix}$, $m_{367}=\begin{pmatrix}1&0&1\\1&0&1\\1&1&0\end{pmatrix}$, $m_{368}=\begin{pmatrix}1&0&1\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{369}=\begin{pmatrix}1&0&1\\1&1&0\\0&0&0\end{pmatrix}$, $m_{370}=\begin{pmatrix}1&0&1\\1&1&0\\0&0&1\end{pmatrix}$, $m_{371}=\begin{pmatrix}1&0&1\\1&1&0\\0&1&0\end{pmatrix}$, $m_{372}=\begin{pmatrix}1&0&1\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{373}=\begin{pmatrix}1&0&1\\1&1&0\\1&0&0\end{pmatrix}$, $m_{374}=\begin{pmatrix}1&0&1\\1&1&0\\1&0&1\end{pmatrix}$, $m_{375}=\begin{pmatrix}1&0&1\\1&1&0\\1&1&0\end{pmatrix}$, $m_{376}=\begin{pmatrix}1&0&1\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{377}=\begin{pmatrix}1&0&1\\1&1&1\\0&0&0\end{pmatrix}$, $m_{378}=\begin{pmatrix}1&0&1\\1&1&1\\0&0&1\end{pmatrix}$, $m_{379}=\begin{pmatrix}1&0&1\\1&1&1\\0&1&0\end{pmatrix}$, $m_{380}=\begin{pmatrix}1&0&1\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{381}=\begin{pmatrix}1&0&1\\1&1&1\\1&0&0\end{pmatrix}$, $m_{382}=\begin{pmatrix}1&0&1\\1&1&1\\1&0&1\end{pmatrix}$, $m_{383}=\begin{pmatrix}1&0&1\\1&1&1\\1&1&0\end{pmatrix}$, $m_{384}=\begin{pmatrix}1&0&1\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{385}=\begin{pmatrix}1&1&0\\0&0&0\\0&0&0\end{pmatrix}$, $m_{386}=\begin{pmatrix}1&1&0\\0&0&0\\0&0&1\end{pmatrix}$, $m_{387}=\begin{pmatrix}1&1&0\\0&0&0\\0&1&0\end{pmatrix}$, $m_{388}=\begin{pmatrix}1&1&0\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{389}=\begin{pmatrix}1&1&0\\0&0&0\\1&0&0\end{pmatrix}$, $m_{390}=\begin{pmatrix}1&1&0\\0&0&0\\1&0&1\end{pmatrix}$, $m_{391}=\begin{pmatrix}1&1&0\\0&0&0\\1&1&0\end{pmatrix}$, $m_{392}=\begin{pmatrix}1&1&0\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{393}=\begin{pmatrix}1&1&0\\0&0&1\\0&0&0\end{pmatrix}$, $m_{394}=\begin{pmatrix}1&1&0\\0&0&1\\0&0&1\end{pmatrix}$, $m_{395}=\begin{pmatrix}1&1&0\\0&0&1\\0&1&0\end{pmatrix}$, $m_{396}=\begin{pmatrix}1&1&0\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{397}=\begin{pmatrix}1&1&0\\0&0&1\\1&0&0\end{pmatrix}$, $m_{398}=\begin{pmatrix}1&1&0\\0&0&1\\1&0&1\end{pmatrix}$, $m_{399}=\begin{pmatrix}1&1&0\\0&0&1\\1&1&0\end{pmatrix}$, $m_{400}=\begin{pmatrix}1&1&0\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{401}=\begin{pmatrix}1&1&0\\0&1&0\\0&0&0\end{pmatrix}$, $m_{402}=\begin{pmatrix}1&1&0\\0&1&0\\0&0&1\end{pmatrix}$, $m_{403}=\begin{pmatrix}1&1&0\\0&1&0\\0&1&0\end{pmatrix}$, $m_{404}=\begin{pmatrix}1&1&0\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{405}=\begin{pmatrix}1&1&0\\0&1&0\\1&0&0\end{pmatrix}$, $m_{406}=\begin{pmatrix}1&1&0\\0&1&0\\1&0&1\end{pmatrix}$, $m_{407}=\begin{pmatrix}1&1&0\\0&1&0\\1&1&0\end{pmatrix}$, $m_{408}=\begin{pmatrix}1&1&0\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{409}=\begin{pmatrix}1&1&0\\0&1&1\\0&0&0\end{pmatrix}$, $m_{410}=\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}$, $m_{411}=\begin{pmatrix}1&1&0\\0&1&1\\0&1&0\end{pmatrix}$, $m_{412}=\begin{pmatrix}1&1&0\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{413}=\begin{pmatrix}1&1&0\\0&1&1\\1&0&0\end{pmatrix}$, $m_{414}=\begin{pmatrix}1&1&0\\0&1&1\\1&0&1\end{pmatrix}$, $m_{415}=\begin{pmatrix}1&1&0\\0&1&1\\1&1&0\end{pmatrix}$, $m_{416}=\begin{pmatrix}1&1&0\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{417}=\begin{pmatrix}1&1&0\\1&0&0\\0&0&0\end{pmatrix}$, $m_{418}=\begin{pmatrix}1&1&0\\1&0&0\\0&0&1\end{pmatrix}$, $m_{419}=\begin{pmatrix}1&1&0\\1&0&0\\0&1&0\end{pmatrix}$, $m_{420}=\begin{pmatrix}1&1&0\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{421}=\begin{pmatrix}1&1&0\\1&0&0\\1&0&0\end{pmatrix}$, $m_{422}=\begin{pmatrix}1&1&0\\1&0&0\\1&0&1\end{pmatrix}$, $m_{423}=\begin{pmatrix}1&1&0\\1&0&0\\1&1&0\end{pmatrix}$, $m_{424}=\begin{pmatrix}1&1&0\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{425}=\begin{pmatrix}1&1&0\\1&0&1\\0&0&0\end{pmatrix}$, $m_{426}=\begin{pmatrix}1&1&0\\1&0&1\\0&0&1\end{pmatrix}$, $m_{427}=\begin{pmatrix}1&1&0\\1&0&1\\0&1&0\end{pmatrix}$, $m_{428}=\begin{pmatrix}1&1&0\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{429}=\begin{pmatrix}1&1&0\\1&0&1\\1&0&0\end{pmatrix}$, $m_{430}=\begin{pmatrix}1&1&0\\1&0&1\\1&0&1\end{pmatrix}$, $m_{431}=\begin{pmatrix}1&1&0\\1&0&1\\1&1&0\end{pmatrix}$, $m_{432}=\begin{pmatrix}1&1&0\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{433}=\begin{pmatrix}1&1&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{434}=\begin{pmatrix}1&1&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{435}=\begin{pmatrix}1&1&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{436}=\begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{437}=\begin{pmatrix}1&1&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{438}=\begin{pmatrix}1&1&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{439}=\begin{pmatrix}1&1&9\\1&1&1\\1&1&0\end{pmatrix}$, $m_{440}=\begin{pmatrix}1&1&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{441}=\begin{pmatrix}1&1&0\\1&1&1\\0&0&0\end{pmatrix}$, $m_{442}=\begin{pmatrix}1&1&0\\1&1&1\\0&0&1\end{pmatrix}$, $m_{443}=\begin{pmatrix}1&1&0\\1&1&1\\0&1&0\end{pmatrix}$, $m_{444}=\begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{445}=\begin{pmatrix}1&1&0\\1&1&1\\1&0&0\end{pmatrix}$, $m_{446}=\begin{pmatrix}1&1&0\\1&1&1\\1&0&1\end{pmatrix}$, $m_{447}=\begin{pmatrix}1&1&0\\1&1&1\\1&1&0\end{pmatrix}$, $m_{448}=\begin{pmatrix}1&1&0\\1&1&1\\1&1&1\end{pmatrix}$,

$m_{449}=\begin{pmatrix}1&1&1\\0&0&0\\0&0&0\end{pmatrix}$, $m_{450}=\begin{pmatrix}1&1&1\\0&0&0\\0&0&1\end{pmatrix}$, $m_{451}=\begin{pmatrix}1&1&1\\0&0&0\\0&1&0\end{pmatrix}$, $m_{452}=\begin{pmatrix}1&1&1\\0&0&0\\0&1&1\end{pmatrix}$,

$m_{453}=\begin{pmatrix}1&1&1\\0&0&0\\1&0&0\end{pmatrix}$, $m_{454}=\begin{pmatrix}1&1&1\\0&0&0\\1&0&1\end{pmatrix}$, $m_{455}=\begin{pmatrix}1&1&1\\0&0&0\\1&1&0\end{pmatrix}$, $m_{456}=\begin{pmatrix}1&1&1\\0&0&0\\1&1&1\end{pmatrix}$,

$m_{457}=\begin{pmatrix}1&1&1\\0&0&1\\0&0&0\end{pmatrix}$, $m_{458}=\begin{pmatrix}1&1&1\\0&0&1\\0&0&1\end{pmatrix}$, $m_{459}=\begin{pmatrix}1&1&1\\0&0&1\\0&1&0\end{pmatrix}$, $m_{460}=\begin{pmatrix}1&1&1\\0&0&1\\0&1&1\end{pmatrix}$,

$m_{461}=\begin{pmatrix}1&1&1\\0&0&1\\1&0&0\end{pmatrix}$, $m_{462}=\begin{pmatrix}1&1&1\\0&0&1\\1&0&1\end{pmatrix}$, $m_{463}=\begin{pmatrix}1&1&1\\0&0&1\\1&1&0\end{pmatrix}$, $m_{464}=\begin{pmatrix}1&1&1\\0&0&1\\1&1&1\end{pmatrix}$,

$m_{465}=\begin{pmatrix}1&1&1\\0&1&0\\0&0&0\end{pmatrix}$, $m_{466}=\begin{pmatrix}1&1&1\\0&1&0\\0&0&1\end{pmatrix}$, $m_{467}=\begin{pmatrix}1&1&1\\0&1&0\\0&1&0\end{pmatrix}$, $m_{468}=\begin{pmatrix}1&1&1\\0&1&0\\0&1&1\end{pmatrix}$,

$m_{469}=\begin{pmatrix}1&1&1\\0&1&0\\1&0&0\end{pmatrix}$, $m_{470}=\begin{pmatrix}1&1&1\\0&1&0\\1&0&1\end{pmatrix}$, $m_{471}=\begin{pmatrix}1&1&1\\0&1&0\\1&1&0\end{pmatrix}$, $m_{472}=\begin{pmatrix}1&1&1\\0&1&0\\1&1&1\end{pmatrix}$,

$m_{473}=\begin{pmatrix}1&1&1\\0&1&1\\0&0&0\end{pmatrix}$, $m_{474}=\begin{pmatrix}1&1&1\\0&1&1\\0&0&1\end{pmatrix}$, $m_{475}=\begin{pmatrix}1&1&1\\0&1&1\\0&1&0\end{pmatrix}$, $m_{476}=\begin{pmatrix}1&1&1\\0&1&1\\0&1&1\end{pmatrix}$,

$m_{477}=\begin{pmatrix}1&1&1\\0&1&1\\1&0&0\end{pmatrix}$, $m_{478}=\begin{pmatrix}1&1&1\\0&1&1\\1&0&1\end{pmatrix}$, $m_{479}=\begin{pmatrix}1&1&1\\0&1&1\\1&1&0\end{pmatrix}$, $m_{480}=\begin{pmatrix}1&1&1\\0&1&1\\1&1&1\end{pmatrix}$,

$m_{481}=\begin{pmatrix}1&1&1\\1&0&0\\0&0&0\end{pmatrix}$, $m_{482}=\begin{pmatrix}1&1&1\\1&0&0\\0&0&1\end{pmatrix}$, $m_{483}=\begin{pmatrix}1&1&1\\1&0&0\\0&1&0\end{pmatrix}$, $m_{484}=\begin{pmatrix}1&1&1\\1&0&0\\0&1&1\end{pmatrix}$,

$m_{485}=\begin{pmatrix}1&1&1\\1&0&0\\1&0&0\end{pmatrix}$, $m_{486}=\begin{pmatrix}1&1&1\\1&0&0\\1&0&1\end{pmatrix}$, $m_{487}=\begin{pmatrix}1&1&1\\1&0&0\\1&1&0\end{pmatrix}$, $m_{488}=\begin{pmatrix}1&1&1\\1&0&0\\1&1&1\end{pmatrix}$,

$m_{489}=\begin{pmatrix}1&1&1\\1&0&1\\0&0&0\end{pmatrix}$, $m_{490}=\begin{pmatrix}1&1&1\\1&0&1\\0&0&1\end{pmatrix}$, $m_{491}=\begin{pmatrix}1&1&1\\1&0&1\\0&1&0\end{pmatrix}$, $m_{492}=\begin{pmatrix}1&1&1\\1&0&1\\0&1&1\end{pmatrix}$,

$m_{493}=\begin{pmatrix}1&1&1\\1&0&1\\1&0&0\end{pmatrix}$, $m_{494}=\begin{pmatrix}1&1&1\\1&0&1\\1&0&1\end{pmatrix}$, $m_{495}=\begin{pmatrix}1&1&1\\1&0&1\\1&1&0\end{pmatrix}$, $m_{496}=\begin{pmatrix}1&1&1\\1&0&1\\1&1&1\end{pmatrix}$,

$m_{497}=\begin{pmatrix}1&1&1\\1&1&0\\0&0&0\end{pmatrix}$, $m_{498}=\begin{pmatrix}1&1&1\\1&1&0\\0&0&1\end{pmatrix}$, $m_{499}=\begin{pmatrix}1&1&1\\1&1&0\\0&1&0\end{pmatrix}$, $m_{500}=\begin{pmatrix}1&1&1\\1&1&0\\0&1&1\end{pmatrix}$,

$m_{501}=\begin{pmatrix}1&1&1\\1&1&0\\1&0&0\end{pmatrix}$, $m_{502}=\begin{pmatrix}1&1&1\\1&1&0\\1&0&1\end{pmatrix}$, $m_{503}=\begin{pmatrix}1&1&1\\1&1&0\\1&1&0\end{pmatrix}$, $m_{504}=\begin{pmatrix}1&1&1\\1&1&0\\1&1&1\end{pmatrix}$,

$m_{505}=\begin{pmatrix}1&1&1\\1&1&1\\0&0&0\end{pmatrix}$, $m_{506}=\begin{pmatrix}1&1&1\\1&1&1\\0&0&1\end{pmatrix}$, $m_{507}=\begin{pmatrix}1&1&1\\1&1&1\\0&1&0\end{pmatrix}$, $m_{508}=\begin{pmatrix}1&1&1\\1&1&1\\0&1&1\end{pmatrix}$,

$m_{509}=\begin{pmatrix}1&1&1\\1&1&1\\1&0&0\end{pmatrix}$, $m_{510}=\begin{pmatrix}1&1&1\\1&1&1\\1&0&1\end{pmatrix}$, $m_{511}=\begin{pmatrix}1&1&1\\1&1&1\\1&1&0\end{pmatrix}$, $m_{512}=\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}$.